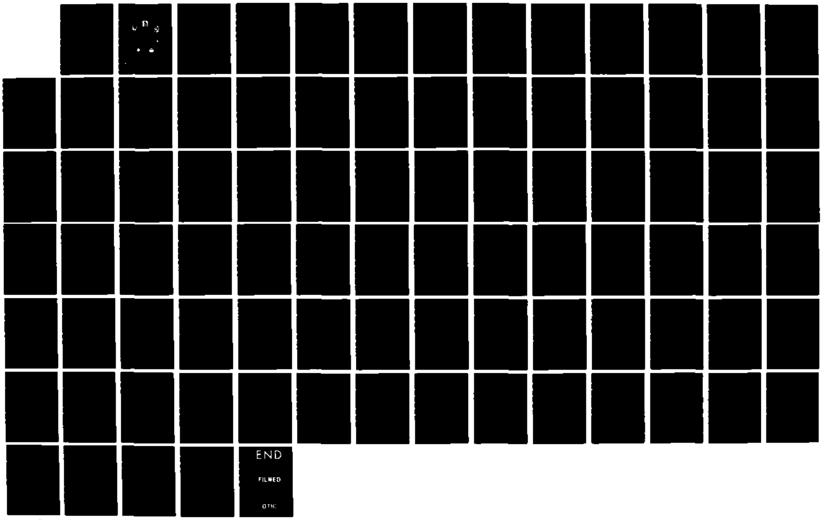
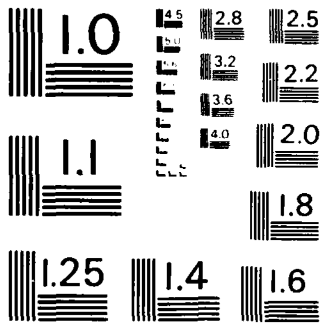
AD-A160 863 ARTIFICIAL INTELLIGENCE IN MAINTENANCE: SYNTHESIS OF TECHNICAL ISSUES..(U) DENVER RESEARCH INST CO SOCIAL SYSTEMS RESEARCH AND EVALUATIO.. J J RICHARDSON ET AL. 1/1

UNCLASSIFIED OCT 85 AFHRL-TR-85-7 F33615-82-C-0013 F/G 5/1 NL




1.0
4.5
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

AFHRL-TR-85-7 October 1985

ARTIFICIAL INTELLIGENCE IN MAINTENANCE: SYNTHESIS OF TECHNICAL ISSUES

JOINT SERVICES WORKING GROUP ON ARTIFICIAL INTELLIGENCE IN MAINTENANCE

Approved for public release; distribution unlimited.

AIR FORCE SYSTEMS COMMAND
AIR FORCE HUMAN RESOURCES LABORATORY
BROOKS AIR FORCE BASE, TEXAS 78235

NOTICE

This report has been reviewed and is approved for publication.

GERALD WALKER
Contract Monitor

JOSEPH Y. YASUTAKE, Technical Director
Training Systems Division

DENNIS W. JARVI, Colonel, USAF
Commander

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY 2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFHRL-TR-85-7
6a. NAME OF PERFORMING ORGANIZATION Denver Research Institute 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Training Systems Division
6c. ADDRESS (City, State, and ZIP Code) University of Denver Denver, Colorado 80208	7b. ADDRESS (City, State, and ZIP Code) Air Force Human Resources Laboratory Lowry Air Force Base, Colorado 80230-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Air Force Human Resources Laboratory 8b. OFFICE SYMBOL (If applicable) HQ AFHRL	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F33615-82-C-0013
8c. ADDRESS (City, State, and ZIP Code) Brooks Air Force Base, Texas 78235-5601	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO: 62205F PROJECT NO: 1121 TASK NO: 09 WORK UNIT ACCESSION NO: 15

11. TITLE (Include Security Classification)
Artificial Intelligence in Maintenance: Synthesis of Technical Issues

12. PERSONAL AUTHOR(S)
Richardson, J. Jeffrey; Keller, Robert A.; Maxion, Roy A.; Polson, Peter G.; and DeJong, Kenneth A.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Interim	FROM Jan 84 TO Jan 85	October 1985	86

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
artificial intelligence, expert systems, maintenance, automatic test equipment, fault diagnosis, training, built-in-test, integrated diagnostics

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report develops and substantiates research and development recommendations for the use of artificial intelligence in improving military hardware maintenance. The genesis of this report was the October 1983 Artificial Intelligence in Maintenance Joint Services Workshop, the proceedings of which have been published by the Air Force Human Resources Laboratory (AFHRL-TR-84-25). The first chapter presents an executive summary and recommendations. The second provides background material on the maintenance problem and artificial intelligence theory and practice. The remaining three chapters present in-depth analyses of integral aspects of maintenance: the automated management of hardware failure, the development and use of human resources, and the larger organizational context of maintenance.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Nancy A. Perrigo, Chief, STINFO Office	22b. TELEPHONE (Include Area Code): (512) 536-3877 22c. OFFICE SYMBOL: AFHRL/TSR

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

ARTIFICIAL INTELLIGENCE IN MAINTENANCE:
SYNTHESIS OF TECHNICAL ISSUES

Prepared by

J. Jeffrey Richardson
Robert A. Keller

Social Systems Research and Evaluation Division
Denver Research Institute
University of Denver
Denver, Colorado 80208

Roy A. Maxion
Peter G. Polson

Institute of Cognitive Science
University of Colorado
Boulder, Colorado 80309

Kenneth A. DeJong

Navy Center for Applied Research
in Artificial Intelligence
Naval Research Laboratory
Washington, DC 20375

A-1

Prepared for

TRAINING SYSTEMS DIVISION
Air Force Human Resources Laboratory
Lowry Air Force Base, Colorado 80230-5000

October 1985

SUMMARY

The objective of this effort was to synthesize the technical issues involved in applying artificial intelligence (AI) to military maintenance systems. The present maintenance situation has many characteristic shortcomings which threaten the services' operational readiness. These shortcomings in the areas of acquisition, technical documentation, training, personnel, logistics, and automatic test equipment are profiled. In addition to the current problems, future trends such as increasing system complexity, diminishing personnel resources, and changing operational scenarios indicate that maintenance challenges of the future will be even more severe.

The science and technology of AI is defined, and how it can help minimize the impact of malfunction on operational readiness is discussed. The principal subdisciplines of AI (e.g., expert systems, problem solving, planning, and natural language understanding) are presented as well as the larger systems engineering issues. In a chapter devoted to automated systems for managing hardware failures, the components of the failure cycle (detection, diagnosis, and repair) are described in tandem with machine approaches and applicable AI methodology.

In this report, effective improvement in military maintenance is viewed to be dependent not only on automated systems but also on the development of human resources and the organizational context of maintenance. Evidence and information are provided to support the recommendation that it is possible to build more effective and less costly automated diagnostic systems only if these systems exploit human problem-solving capabilities. Four hypothetical examples of advanced systems and a comparison of human vs. machine strengths and weaknesses as problem solvers are outlined.

Five research and development recommendations for the use of AI in maintenance conclude that (1) there is a good match between the need for improved maintenance and the emerging science of AI, (2) AI research should be guided by a policy of integrated diagnostics, (3) field evaluations of AI applications should focus on organizational impact as well as technical issues, (4) programs should be targeted at both fielded systems and systems under development, (5) basic research should investigate cooperative human-machine device diagnosis problem solving and the coordination of the specification- and symptom-based approaches.

PREFACE

The genesis of this report was the October 1983 Joint Services Workshop on Artificial Intelligence in Maintenance. The proceedings of that workshop have been published by the Air Force Human Resources Laboratory (Technical Report AFHRL-TR-84-25).

This document was developed from January 1984 through January 1985 by the Denver Research Institute, J. Jeffrey Richardson, Principal Investigator. The work was sponsored by the Air Force Human Resources Laboratory under contract F33615-82-C-0013, and the contract monitors were Major Hugh L. Burns, and Brian E. Dallman.

The authors would like to thank the Air Force Human Resources Laboratory, specifically Hugh Burns and Brian Dallman, for their support, direction, and encouragement in the development of this report and the Joint Services Workshop that preceded it. Gratitude is also extended to Bonita L. Moul, whose editorial assistance helped pull the work of five authors into a coherent whole.

TABLE OF CONTENTS

I. EXECUTIVE SUMMARY AND RECOMMENDATIONS

This chapter summarizes the synthesis of technical issues involved in the application of artificial intelligence (AI) to military maintenance systems. It proposes an approach to research, development, and application which integrates automated fault handling technology, human resources, and organizational support.

Statement of the Problem

The scope of maintenance far exceeds its core activities of detection, diagnosis, and repair of faults. Maintenance concerns actually begin with system design: design for reliability, maintainability, and testability. In addition to design, maintenance concerns include acquisition, built-in and automatic test, technical documentation, maintenance training, personnel, and logistics. Shortcomings in each of these specific areas of maintenance exist today and because of increasing systems complexity, diminishing personnel resources, and changing operational scenarios, the maintenance challenges to be faced in the future are even more severe.

The literature has characterized the current maintenance shortcomings as follows: Electronic data systems, including design, engineering, manufacturing, operations, maintenance, and training, are insufficiently integrated. Built-in and automatic test systems have high false alarm, false removal, and manual test rates which result in unnecessary maintenance activity. Test program sets for automatic test equipment are costly to generate, high in number, and run inflexible, lengthy test sequences. Paper-based technical documentation is physically bulky and difficult to use. The low priority of training activities results in inadequate numbers of trained instructors and up-to-date equipment. The quantity and quality of the labor pool is decreasing, civilian opportunities dampen reenlistment, and there is no method for systematically capturing the experiential knowledge of senior technicians before they leave the services. In sum, severe problems are said to exist throughout the scope of maintenance activities.

Why AI Can Help

Artificial intelligence is the science and technology of reproducing human-level intellectual competence with machines. That is, AI is the practice of building process models of intellectual activity that can be run on a computer. The main intellectual activities of interest include problem solving, learning, and natural language processing. These activities generally involve complexity (designing a bridge), uncertainty (deciding whether to buy or sell on today's stock market), or ambiguity ("John said Jack said he went to the store."). All of these activities involve knowledge and the manipulation of knowlege in achieving a goal. Taking problem solving as an example, the basic AI approach is to create a space

of all possible sequences of allowable problem-solving steps and then search this space for a sequence that leads to a valid solution. This search is neither random nor exhaustive; it is guided in order to limit the number of potential solutions considered. This example illustrates two central issues of artificial intelligence: representation of knowledge and methods of controlling a search. In general the objective is to arrive at a good solution most of the time as opposed to the best solution all of the time.

AI can help in solving modern maintenance problems if computer-based systems can do more of the human-level intellectual tasks required in maintenance. More fundamentally, AI can help because it is interdisciplinary, sharing much of two principal disciplines of importance in maintenance: psychology and computer science.

Meeting the Objective

The maintenance objective is to minimize the impact of malfunction on operational readiness. Rapid progress toward this objective can be reached through a coordinated research and development (R&D) program targeted at the broad scope of maintenance activities. Therefore, a program of AI R&D in maintenance will have the greatest impact when it recognizes and reinforces maintenance interrelationships--this is the policy of Integrated Diagnostics (National Security Industrial Association, 1984a). The remainder of this chapter describes an AI R&D program in maintenance, presented within the framework of integrated diagnostics.

New System or Old?

There are two alternatives in choosing a maintenance system to provide an environment for R&D: fielded systems and systems under development. Both environments provide important niches for investigation of AI potential. Yet, neither gives access to the whole picture, which is design and support considerations and their interrelationships.

Fielded systems offer experience, data, and a stable operational environment that a system under development cannot provide. The AI R&D in support of a fielded system will also yield results first and with less risk. The results will be available for dissemination before those for a new system because of prime system development lag time, and risk is smaller because it is possible to take advantage of accumulated maintenance experience, a source of knowledge unavailable in designing support for a new system.

On the other hand, a system under development offers the opportunity to bring maintenance concerns and the technology for meeting them directly into the systems design phase. Nothing, including AI, can remediate a poorly designed system--maintenance problems stemming from design must simply be tolerated. Thus, the most leverage in improving operational readiness is available during the systems design phase.

Due to the distinct advantages provided by both old and new systems, an AI R&D program should be initiated to investigate each. R&D activities with old and new systems phase nicely because the R&D for fielded systems provides a technology base for new systems developed with innovative AI approaches to design. Both approaches are recommended.

Redeveloping Maintenance Support Systems

Any adequate solution to the maintenance problem must capitalize on the interrelationships between automated fault handling systems, trained personnel, and organizational support. Therefore, the top level goal in this report envisions an AI-based diagnostic system in an integrated context. A description of the structure and function of the equipment to be maintained is the basis for integration.

This description is the principal source of information needed to drive a form of diagnostic reasoning compatible with both AI and human approaches to diagnosis. Since human expertise has always been, and probably always will be, needed to augment or complement automated diagnostics, the human-computer interface is an important aspect of such a diagnostic system. The human-computer interface design centers on means of generating explanations for the user regarding diagnostic information processing. In this way, the user can better monitor the automated diagnostic processing and take over when necessary. Also, through more structured tutorial interaction, the system can serve to increase the user's competency.

At the organizational level, mean time between failure, test cost, and other data from maintenance information systems is used by the diagnostic system in controlling search. The maintenance information systems should be designed to facilitate the forward and reverse flow of information between individual maintenance events and aggregate data at the organizational level. With this overview in mind, specific issues relevant to each facet of the system (automated systems for managing hardware failures, human resources development and use, and the larger context of maintenance systems) are presented below.

Automated Systems for Managing Hardware Failures

The failure cycle is a sequence of events which forms the context for maintenance activity. When a fault occurs, it must first be detected. This is the main function of built-in test equipment. Then the fault must be diagnosed or isolated, the main function of automatic, off-line test equipment. Then, based on a known source of failure, system recovery must be made. The source of failure may be replaced or the system reconfigured to compensate for the failure. Finally, to begin the cycle over again, there is a possibility of predicting a fault in advance based on real time or background analysis.

Fault Detection

Most fault detection technology is incorporated directly into the hardware of the system under test. On-line monitoring is biased by its mission toward a high false alarm rate. This rate can be excessively high. Many approaches are currently used in the built-in test community to reduce the false alarm rate of built-in test. These include duplication, error detection codes, watchdog timers, and consistency and capability checks. Expert systems approaches to built-in test would add to this technology in two ways. First, since this is a software approach, the performance of the system can be improved without needing to make hardware changes. Second, either human or machine-based analysis of the system performance can be used to add new rules to the expert system's rule base to increase built-in test performance.

Fault Diagnosis

Diagnosis is the process of isolating a fault through repeatedly making measurements, computing their entailments, and selecting the next test to make. (The "next test" is selected on the basis of maximizing information gain per unit cost.) There are two fundamental approaches to diagnosis: symptom-based and specification-based.

The symptom-based approach, often termed shallow reasoning, solves diagnostic problems by manipulating a set of associations between symptoms and faults. With this approach, the associations between symptoms and faults are heuristic in nature and based more on experience than on reasoned causal derivation. This approach may employ tactics for capturing the times and locations of observed errors. This aspect is appealing because it bears so much similarity to what a technician might observe in a failing system.

The symptom-based approach is completely device dependent. It can, however, easily handle symptom-fault pairings that defy the specification-based approach. Many technology demonstrations are based on this approach, where the diagnostic rules (empirical associations) are developed by a knowledge engineer working in conjunction with a subject-matter expert. This process, called knowledge acquisition, is recognized as a bottleneck in the expert systems development process. In spite of the knowledge acquisition bottleneck, expert systems based on empirical associations are applicable in cases where human judgement is the principal source of knowledge, for example, at organizational maintenance level.

The specification-based approach, often termed deep reasoning, solves diagnostic problems by reasoning from the structure and behavior of the device. The structure is a description of the connectivity or dependency of its components. The behavior is a description of the input-output behavior of each component. Using these descriptions only, the composite behavior of the system can be derived through the propagation of individual component behavior through the connectivity network. This propagation is constrained by applicable network laws, such as Ohm's and Kirchoff's Laws. Often multiple possible composite

behaviors are generated through this causal propagation. Knowledge of the device's intended purpose or function can be used to rule out incorrect derivations of composite behavior.

Specification-based diagnosis is the prevalent approach of AI research in this field. It holds the ultimate promise of developing diagnostic systems that require the absolute minimum device dependent knowledge (a description of its structure). In this way, this approach maximizes the generality and robustness of a diagnostic system. However, this approach is as difficult to achieve on a practical scale as it is ambitious. The fault coverage of specification-based diagnostic systems is limited by the completeness and accuracy of the structural description on which it is based. Components may behave in ways that are not modeled. Alternative paths of causality may exist besides the ones specified in the component interconnections. Or, in the cases of field work-arounds or temporary fixes, the specification of the device will simply be inaccurate in places.

The symptom- and specification-based approaches are not separate, independent, or distinct. For example, there must be a causal explanation for every empirical fact, but often these connections are hard to make. Moreover, as people become familiar with and begin to recognize recurring symptom-fault associations, they will prefer to use these rather than resorting to reasoning "from first principles." Repeated specification-based derivations of a given symptom-fault implication will (routinely, in human performance, or by design, in machines with a learning capability) be replaced with simple associations that skip (or compile out) the intermediary steps in a causal argument.

In intelligent human behavior both approaches to diagnosis are employed. The symptom-based approach is preferred, because it requires less reasoning than does the specification-based approach, which is used only when the other fails. In general, as humans acquire expertise, the reasoning process grows and develops from a goal-directed problem-solving approach (the specification-based approach) to a pattern-directed, associative approach (the symptom-based approach).

To illustrate the applicability of both diagnostic approaches in military maintenance, consider the following examples drawn from the three repair levels of the modern maintenance system: organizational, intermediate, and depot. At the organizational maintenance level, cumulative experience provides powerful heuristics, rules of thumb which shortcut more formal approaches. This type of expertise is a perfect match for the rule-based expert system. At the intermediate and depot levels of maintenance, additional sources of knowledge, such as circuit topology or circuit dependencies, become more useful. The process of entering rules to capture this type of knowledge is highly inefficient. This information is deriveable from computer-aided design (CAD) data, or can be developed by technicians from circuit diagrams.

AI R&D in the area of diagnosis should not focus exclusively on either one of these approaches. In addition to further development of a technology for each approach, attention should be paid to how these two approaches can be integrated. In fact, this integration is key to progress in machine learning in this area.

To date, diagnostic expert systems do not learn. Expert systems and machine learning are separate subfields of AI. The expert systems field has enjoyed commercial success and visibility ahead of machine learning because performance is an easier problem to solve than learning. It is an important goal for expert systems research to develop systems that learn. A major position on machine (and human) learning is that learning is a slow, incremental process of expanding a highly organized knowledge base. Issues involve what representations of knowledge and what processes (e.g., the combination and differentiation of rules) support the building of new knowledge.

Fault Recovery

The basic means of fault recovery are switching to redundant systems, repairing or replacing faulted systems, and reconfiguring overall systems to compensate for a fault. In the area of reconfiguring systems, expert systems technology is being applied to reconfigure digital flight control systems. Existing work in configuring computer systems might be applicable to the related task of reconfiguring systems. Reconfiguration depends on having a model of the function and structure of the system, a scheme for ordering the importance of various functions, an ability to plan sequences of actions, and a knowledge of when no compensating strategy will provide adequate recovery.

Often it is not possible or there is no time available to reconfigure a system. In such cases, information needs to be developed to make an operational decision, as opposed to a maintenance decision, regarding what the degraded system's performance capabilities currently are and how this impacts the mission. These decisions involve a wide range of information, uncertainty, and experienced human judgement; in other words, a good expert systems application.

Fault Prediction

Anticipation of incipient faults depends on pattern recognition and trend analysis based on a log of parametric data. Systems existing today, for example in the M-1 tank or B-1B bomber, can monitor and log such data. Taking these data and turning them into knowledge (that is, fault predictions based on these data) is another application area of AI. Relevant AI disciplines would be expert systems (capturing the knowledge of experienced technicians who can interpret such data) and machine learning (supporting the recognition of new fault signatures). In order for a fault prediction system to increase its competence through learning, further basic research needs to be conducted in causal models of physical systems and in machine learning. The goal would be a fault prediction system that could improve its competence over time, based on experience.

Explanation for Designers

There is a need for explanation to support the development and maintenance of AI systems. While complete sources of knowledge, such as a

description of device structure derived from computer-aided engineering data, may be entered automatically, debugging, tuning, updating, verifying, validating, and maintaining a system that is based on multiple sources of knowlege will require an efficient and effective method of interface with the system builder. The more principled the method of entering knowledge into an expert system and the more explicit in form that knowledge is, the more accessible it will be to the system builder.

Developing and Using Human Resources

The Need for Trained Technical Personnel

Advances in equipment design, automated failure prediction, detection, diagnosis, and recovery will tend to decrease the requirement for trained technical personnel. With advanced automated systems, human involvement tends to be limited to unskilled or semi-skilled activities. The human acts as sensor and manipulator, carrying out computer-generated instructions to check test points, remove and replace modules, etc. However, skilled technical personnel will continue to be a vital part of the maintenance system for the following reasons:

1. Automated diagnostics will always be imperfect to some degree; the human is diagnostician of last resort.

2. Automated systems will at times be unavailable when they are needed.

3. Human validation, verification, and suggestion for improvement of automated systems is a vital part of the maintenance system.

4. Human dignity suffers, reducing morale and motivation, when human cognitive capabilities are underutilized.

These points establish the need for trained technical personnel in modern maintenance environments, regardless of the level of sophistication of automated systems. Therefore, automated systems ought to be designed to support the development, maintenance, and use of human expertise. Such a system is defined as cooperative human-computer problem solving.

The development, maintenance, and use of human expertise can be accomplished by providing training and cognitively engaging activity to the technician in the context of his or her job, at appropriate levels of detail. From the vantage of integrated systems, the traditionally separate support technologies of training and technical documentation should be integrated with each other and with automated fault-handling systems.

The Potential Range of Integrated Job Aiding and Training Systems

The following examples illustrate a potential range of integrated job aiding and training systems. Within these examples, the degree of human involvement in the diagnostic task is varied. The first example is at one extreme in which the human is employed only as sensor and manipulator and follows instructions from the computer in how to perform. All diagnostic reasoning is carried out by the automated system; no human intervention in the diagnostic reasoning process is required or anticipated. Therefore, only semi-skilled personnel are needed. This system makes the unreasonable assumption that complete 100 percent fault isolation can be effected through automated means.

A more realistic example retains the basic features of the above approach, except provision is made for smoothly passing a diagnostic problem to an expert human diagnostician when the automated system is unable to isolate a fault. AI implications for this scenario are that the automated system knows when it has failed and can explain to the human what knowledge had been developed thus far in the course of the diagnosis. These are both challenging issues within current AI research. The main personnel and training implication is that sophisticated diagnostic expert technicians must be supplied to the system. Since the machines do all the routine work, no opportunity exists for incremental skills development on the job.

A third example is termed the master-apprentice approach. Here an attempt is made to transition automated diagnostic expertise employed in the above examples to a human apprentice through appropriate on-the-job training and explanation mechanisms. The main AI implication is that a methodology for the development of intelligent tutorial systems must exist, including the ability to base explanations and sequence job experiences on an accurate model of the apprentice's current competencies. Research in this area is maturing, but prescriptive methodologies specifically applicable to maintenance have yet to be developed. The training implications are favorable, in contrast to the above two scenarios, because in the master-apprentice approach there is a means of incrementally developing the advanced human expertise required when automated systems cannot fault isolate.

In the final example, the mixed-initiative human-computer diagnostic system, both the person and the automated diagnostic system are directly involved in diagnostic problem solving. The objective of this system is to maximize overall diagnostic adequacy by effectively combining complementary capabilities of human and machine. This is an extremely difficult problem which little or no applied AI research addresses. This approach is compatible with the previous example, yet extends it: when the apprentice's skills are fully developed, the two work jointly as peers.

Because of the need for human involvement in diagnostic tasks, AI R&D in intelligent maintenance aids should investigate the designs of the last three examples above.

Psychological Issues

In the failure cycle, the greatest need for human involvement is diagnosis. This is also the most difficult area in which to make improvements. Continued psychological research in three areas is necessary to establish the technology needed to build automated diagnostic systems which both exploit human problem-solving skills and help people grow on the job: diagnostic problem solving, skill acquisition, and explanation.

Diagnostic problem solving. There is a strong theoretical foundation for understanding human diagnostic reasoning. The most common form of human diagnostic problem-solving behavior is mediated by direct associations between symptoms and faults, referred to as the shallow or symptom-based model. Deep reasoning is another mode of diagnostic problem solving and involves making inferences about possible faults given first principles, descriptions of function and structure, and information about a particular set of symptoms. While these two general modes of diagnostic reasoning are understood, additional basic research is needed so that the general principles of diagnostic inference applicable across problem areas (electronics, hydraulics, mechanics, etc.) can be identified and the relationship between the two problem-solving approaches (shallow and deep) can be better understood.

In addition to basic research, exploratory development in this area is also required because, while diagnostic problem solving has a good theoretical base, there remains the need to apply this base and develop explicit diagnostic reasoning systems for specific maintenance tasks. These cooperative human-computer problem-solving systems need to possess a breadth and depth of diagnostic competence useful in real maintenance environments. The cooperative systems should recognize, accommodate, and supplement for the failure modes of human diagnostic reasoning documented in psychological literature, such as working memory failures, set and functional fixity, inference failures, and attention to irrelevant information.

Skill acquisition. Issues regarding skill acquisition are vital to the development of competent technical personnel. A paradigm for research in the area of skill acquisition is the development of intelligent tutorial systems (ITS). Advances in ITS hinge on several skills acquisition research issues including: appropriate models of diagnostic reasoning, for both the novice and expert diagnostician; the nature of skill acquisition, e.g., the changes in reasoning which accompany the development of increased competence; the appropriate level of detail at which to model student performance; methods for inferring student competence (which involve the AI topics of plan recognition, learning, and dealing with randomness in behavior); theories of instruction useful in sequencing lessons and which provide guidance on the relative roles of exposition, example, and practice; process theories of how to be a good tutor; means of broadening interaction with the student (such as natural language and graphics input); issues regarding the generation of explanations (when to, how to, and what are the characteristics of useful explanations); and designs of interactive environments upon which to base instructional interactions (such as problem-solving editors or gaming environments).

ITS have been developed which address some of these issues. One relatively mature ITS design is the "black box" system. In this design, the system's expertise is inaccessible for the purposes of explanation since it is represented as circuit simulations or coded algorithms. The student is modeled by a set of issues on which system and student performance are compared. Research continues with ITS incorporating articulate expert modules which, in contrast to the black box system, employ articulate experts able to interact with students in terms of the basic elements which constitute expertise (first principles, heuristic rules, problem-solving strategy, and general knowledge). With articulate expert modules, the student's expertise is modeled as a subset of the expert's full set of rules or as a "buggy" version of the expert's set of rules. The utility of ITS as a component of automated diagnostic systems depends on the extent to which the diagnostic system's knowledge is well principled and accessible. A set of rules that enable an expert system to perform at a given level of expertness is not necessarily a satisfactory basis of instruction. For instruction, a rule set must be explicitly able to support the kind of justification and explanation learners require. The construction of articulate experts specifically useful for teaching is an active area of exploratory development in its infancy.

Explanation for users. For the users of integrated job aiding and training systems, explanation is needed in two contexts: in response to the initiative of the user (for example, the user wonders why he or she is being asked to make a particular measurement) and in the context of instruction, where the system takes the initiative. Explanation is a current issue in the expert systems field. The most prevalent way of providing an explanation is to present canned text associated with the goals and subgoals the expert system is currently pursuing. The adequacy of this approach is minimal, especially if the expert system was developed without structuring the knowledge base in a disciplined way. Not only must the knowledge within the system be properly represented to serve as the basis for explanation, but the way explanations are formulated and delivered (what to say, when to say it, and how to say it) should be responsive to the user's current needs, beliefs, goals, and knowledge. That is, truly adequate explanations require a model of the user.

Personnel Issues

The case has been made that the advent of intelligent maintenance aids will not eliminate the need for trained technical personnel. These aids will reduce the overall need for personnel with intermediate level skills, while retaining a need for highly trained technicians. The issue facing military maintenance organizations is how to sustain a base of highly skilled personnel. Two different approaches are possible: separate careers for semi-skilled and for highly skilled personnel with separate recruitment and selection criteria, or a pipelining approach where indiviudals with aptitude and promise are provided advanced training after an initial tour of duty as semi-skilled technicians.

In either case, the technical requirements for intelligent aids are substantially the same. In each scenario, the aid must be able to provide "how-to" explanations to lesser skilled personnel. In each scenario, the aid must stop work

on a problem when the problem lies beyond its competence and provide a useful summary debriefing of the problem-solving activity that it has performed up to that point. In each scenario, the aid must be able to coach its user. In the separate tier scenario, this coaching is used by the upper tier only, while in the pipeline scenario, it is employed in both tiers.

Because the engineering features of an intelligent maintenance aid are identical for either scenario, the choice between the two scenarios is independent of the aid and of the artificial intelligence technology which supports it. The choice rests on an analysis of the values, constraints, resources, and mission of a particular organization.

Organizational Support

The previous sections discuss automated fault-handling systems and development of an educative link between this base and humans. The current section takes this process one step further and considers AI applications at the level of the organization: how is information fed through the maintenance system, how can plans be made to maximize efficiency and control cost, and how can scarce resources be wisely allocated throughout the maintenance system?

Recall that one of the primary purposes of humans in the diagnostic loop is to evaluate and maintain the quality of the diagnostic system; that is, to serve as a source of information. Therefore, at the organizational support level of an integrated system, it must be possible to incorporate feedback from the field into the diagnostic rule base. Similarly, system updates must be passed back to the field. In addition to diagnostic information are additional kinds of information which include prime system and part number histories, routine maintenance reports, parts inventories and orders, and job schedules. Also of importance is the information flow from operations to maintenance: operator interrogation and debriefing and built-in test and parametric data logs.

AI research areas applicable to information management include knowledge-based systems, natural language, and learning. Knowledge-based systems can provide the type of expertise needed to link together different information sources. For example, actual and projected parts inventories may be stored on distinct and geographically distant systems. These need not be integrated into a single data base running on a single system. Rather, they can be interfaced through a knowledge-based system that is able to access information from each data base.

Natural language technology is important whenever humans need access to information. The same technology employed by AI, which reduces syntactically different but semantically identical statements to identical machine representations, may be as useful in machine-machine communication as it is in human-machine communication.

Finally, machine learning has great applicability in this area because data on which to base learning are most available at the organizational level. Thus, machine learning algorithms could be developed to analyze the contents of various data bases looking for trends or inconsistencies.

The AI field of planning is applicable to resource allocation. Work now underway is addressing job-shop scheduling and interactive planning and execution monitoring.

Recommendations

1. Programs of research, development, and application in artificial intelligence in maintenance are warranted because of the match between need for improved maintenance proficiency and the emerging maturity of AI device diagnosis and intelligent tutorial systems.

2. Applied AI research in maintenance should be guided by Integrated Diagnostics policy. Integration should be achieved with the use of a single representation of device structure, suited for design; automated systems for managing failure; the development and use of human resources; and the organizational context of maintenance.

3. Field evaluations of AI applications to maintenance should focus not only on technical issues but also on the potential organizational impact of the technology.

4. Programs should be targeted at both fielded systems and systems under development.

   4.1 Exploratory development programs for fielded systems:

   - should be conducted at a scale large enough to test the validity of the integrated systems approach and provide a rich enough environment to generate new research issues and findings.

   - should develop tools and methods to build intelligent maintenance aiding systems which:

     - contain a structural model of the device to be maintained

     - contain accumulated field knowledge about the device and its maintenance

     - perform automated diagnosis driven by both of the above two sources of knowledge

- provide instructions for how to carry out sensory and manipulatory activities

- provide explanations of diagnostic activity

- provide tutorial interaction in the context of an on-the-job skills development program for the maintenance of the target device

- accumulate and forward maintenance data to relevant maintenance information systems

4.2 Exploratory development programs for new systems involve issues not addressable in programs of exploratory development for existing systems. Specifically, these issues include technologies for:

- building computer-aided design systems which facilitate the consideration of reliability, testability, and maintainability during the design process

- structuring and formatting design and engineering data so that it may be automatically forwarded through intelligent maintenance aiding systems

- developing maintenance information systems to support the accumulation of performance data for use in intelligent maintenance aiding systems. Performance data should include both machine generated data (built-in test and sensor logs) and human generated data (operator debriefings and maintenance logs).

5. A basic research program in AI applications to maintenance should be initiated to investigate:

- the concept of cooperative, mixed-initiative human-computer problem solving in the area of device diagnosis

- the coordination of specification-based and symptom-based diagnostic problem solving and mechanisms through which diagnostic effectiveness and efficiency can be increased based on the accumulation of maintenance event data

- the conversion of parametric data collected during system operation to knowledge that can be used to predict, detect, diagnose, and recover from incipient malfunctions

II. BACKGROUND

The scope of maintenance far exceeds its core activities of detection, diagnosis, and repair of faults. Maintenance concerns actually begin with system design and involve specifics of diagnostic strategies, job aids, technical documentation, training, personnel, precision measurement, maintenance management, and spares.

The present maintenance situation must be interpreted in the context of the prevailing armed services maintenance concept. The services currently employ a three-tier arrangement that relies heavily on automated test systems. At the organizational level, a fault is isolated through built-in test routines to a line replaceable unit (LRU), a black box that can be removed from a system and replaced with a good one. At the intermediate shop level, the removed LRU is tested on automatic test stands where the fault is further isolated to a specific printed circuit board which is removed and replaced. Finally, at the depot level, the circuit board is tested both manually and with additional automatic test equipment to isolate the faulty replaceable component.

Built-in test (BIT) and automatic test equipment (ATE) are the basic tools of the services' maintenance approach. Their development and use was the necessary response to increased hardware system complexity. But, in spite of and, in certain cases because of, BIT and ATE, serious maintenance problems persist.

Since electronic systems, particularly avionics, impose the greatest demands on maintenance resources, they have provided the focus for both the Joint Services Workshop on Artificial Intelligence in Maintenance (AFHRL, 1984) and this report. In this chapter, background information essential to an understanding of AI in maintenance is presented. The Statement of the Maintenance Problem section involves four ingredients: (a) current problems, (b) future trends, (c) the response of the Department of Defense (DoD) to these problems, and (d) two scenarios which depict maintenance today and hopes for the future. The second section involves ways in which AI can help solve maintenance problems. AI is defined, the subdisciplines of AI having potential applications to maintenance are reviewed, AI systems engineering issues are discussed, and finally, the pragmatics of AI research are described.

Statement of the Maintenance Problem

The literature indicates that the current military maintenance situation has many characteristic shortcomings which may threaten the services' operational readiness (McGrath, 1984). The problems described in this section characterize the current maintenance situation; items are listed without regard to whether or not a contribution can be made by artificial intelligence.

It is also believed that future maintenance challenges will be even more severe due to increasing systems complexity, diminishing personnel resources, and changing operational scenarios (Halff, 1984). These future challenges are outlined following the discussion of the current maintenance environment.

Current Shortcomings

Acquisition.[1] The acquisition process is in need of more standardized procurement requirements related to reliability and maintainability as part of system design. Current maintenance shortcomings which have been cited in this area:

- insufficient integration of electronic data systems--design, engineering, manufacturing, operations, maintenance, and training

- inadequate methods of risk assessment, cost control, and the means to track performance

Built-in and automatic test.[2] Attempts to automate the diagnosis process through BIT and ATE have fallen short of initial expectations. Specifically, the issues which have been discussed:

- high cannot-duplicate rates--intermittent faults, transient faults, and false alarms may comprise as much as 25 percent of all maintenance events

- high manual test rates--often up to 50 percent of hard faults must be isolated manually

- high false removal rates--due to diagnostic error, from 15 to 30 percent of units in the maintenance stream are actually good, accounting for over a third of personnel hours expended on maintenance

- some test programs which have excessively long execution times, large replacement ambiguity groups, are inflexible in test sequencing, and costly to generate

- ATE of extreme bulk (e.g., the intermediate test shop for one F-16 fighter wing requires six C-5As to transport it).

---

[1] Mooney, 1984.

[2] Coppola, 1984; Institute for Defense Analyses, 1981; Lahore, 1984; McGrath, 1984; National Security Industrial Association, 1984a, 1984b; Shumaker, 1984.

Technical documentation.[3] The system of technical documentation as it exists today is paper-based and therefore physically bulky. Other characteristic problems noted in the literature:

- poor coordination and cooperation between the creators of technical documentation and the engineers who designed the system

- inadequate readability and usefulness, including the presentation of information in line with technicians' needs and mental approaches to maintenance problem solving

- insufficient coordination between technical documentation and instructional materials used in residential training

Maintenance training.[4] Some of the problems currently faced in maintenance training are:

- a low priority of training activities which often results in overworked and underqualified instructors who must teach using obsolete equipment

- the trade-off between in-residence and on-the-job training complicated by the need to train technicians to maintain increasingly complex systems

- a growing need to provide basic skills remediation because increasing numbers of recruitable youth have educational and language deficiencies

- an Instructional Systems Development process which needs improvement

- maintenance training simulation with too much focus on physical fidelity--research on physical fidelity needs to be augmented with attention to cognitive processes

- shortcomings in the underlying scientific base of the psychology of learning and instruction

- costly and labor-intensive instructional development and delivery methods--improvements in effectiveness and efficiency are needed

---

[3]Halff, 1984, National Security Industrial Association, 1984b.

[4]Halff, 1984; Montague & Wulfeck, 1984; National Security Industrial Association, 1984b.

Personnel.[5] The availability of personnel at both the entry and skilled levels is diminishing. Furthermore:

- intellectual aptitude has been declining (as measured by Scholastic Aptitude Test scores)

- demand for skilled technicians in the private sector is fierce when the economy is healthy: retention is a serious problem

- there is no method for capturing the experiential knowledge of senior technicians before they leave the services

Logistics.[6] This area includes maintenance information, analysis, and support systems. The problems noted in the literature include:

- insufficient coordination among the various important data bases, such as supply, history, operations, and maintenance scheduling

- an excessive number of spare parts in the maintenance pipeline due to false removals

The shortcomings listed above indicate that there is much unnecessary maintenance activity (Coppola, 1984). This is manifest in maintenance facilities that are overloaded, inflated requirements for spare units, excessive requirements for trained technicians, and limited resources for training. In sum, the current military maintenance situation is characterized by excessive cost, bulk, and a lengthy logistics tail.

Future Trends

In addition to the shortcomings that exist today, there are three future trends which compound maintenance problems and promise to increase the difficulty of supporting weapon systems: (a) continued increases in system complexity; (b) diminishing personnel resources; and (c) operational requirements for the late 1990s and early twenty-first century.

Continued increases in system complexity. Advancing technology is complicating, not simplifying, the maintenance task for modern hardware systems. Technological advances more often enhance functionality than

---

[5]Halff, 1984; Lahore, 1984; McGrath, 1984; National Security Industrial Association, 1984b.

[6]Coppola, 1984; McGrath, 1984; National Security Industrial Association, 1984b.

reliability. Thus, while the field radio of the World War II era had a mean time to failure of approximately 50 hours, the modern field radio, with truly remarkable functionality, has a similar failure rate and is harder to fix (Shumaker, 1984).

Another impact of increased complexity is that ATE is now necessary to support maintenance. The services are preparing the next generation of ATE with the emphasis on standardizing interfaces and architecture (in the Navy, the Consolidated Support System, CSS; in the Army, the Automatic Test Support System, ATSS; and in the Air Force, the Modularized Automatic Test Equipment program, MATE). In addition to new off-line ATE, systems will require increasing amounts of built-in testing. The impact of increased automatic testing on the current shortcomings outlined above is not yet clear.

The volume of maintenance documentation has also soared. A plot of the number of frames or pages of technical documentation for selected Navy aircraft over the past 40 years indicates technical manual size is doubling about every 5 years (Halff, 1984). Paper-based documentation is becoming out of date due to sneer volume alone. To keep pace with increasing complexity, the length of technical training courses has also increased. Thus, the already long logistical support tail for new and technologically advanced systems is becoming longer.

Diminishing personnel resources. Between 1978 and 1990 the pool of 17-year-old males and females will decline by 24 percent (Halff, 1984). This is not an estimate because 1990's 17-year-olds were born in 1973. Not only do existing personnel have to be replaced, the service's total personnel requirement is growing. For example, the Navy's personnel needs will expand when its fleet of 400 grows to 600 ships. Thus, at the very time when more highly skilled people are needed by the military, the supply of young persons of all aptitudes is declining.

In addition, the competition for bright young people is stiff. The ability of the military services to attract such recruits in the open marketplace is often at the mercy of short-term national economic trends. The services cannot rely on counteracting advancing technology's impact on maintenance by recruiting more and brighter maintenance personnel.

Future operational requirements. There are three operational requirements in the services for the late 1990s and early twenty-first century that will affect maintenance (McGrath, 1984). First, the services will be required to sustain intense surges, up to 72 hours in duration. This means maintenance-free operations for at least this period of time will be necessary to sustain high sortie rates. Also, to keep sortie rates up implies a need for high system reliability and maintainability, fault-tolerant or self-repairing systems, and self-reconfiguring systems. Second, there will be small, highly mobile units. This will require logistics command and communication systems, paralleling those for combat operations, to coordinate the logistics support. Finally, the services will mobilize against a more capable threat. Increased system capability and performance are the desired results of the greater system complexity.

Department of Defense Initiatives

The services have been working diligently to address these maintenance problems. Improvements in personnel selection and classification, training, military pay, equipment reliability, technical information systems, job performance aids, and logistics support systems are all responses to the current situation.

The Department of Defense has also established a weapon support and logistics research and development initiative with objectives of technology demonstrations in five areas: automation of technical information, logistics command and control, automated battlefield material handling, automated "parts on demand" manufacturing, and reduced or eliminated intermediate maintenance (McGrath, 1984; National Security Industrial Association, 1984a). All projects conducted under this initiative will contribute to alleviating maintenance problems.

The recognition that all aspects of maintenance, from acquisition to spares, are integrally related is an additional DoD response. By explicitly recognizing this interrelatedness, greater overall improvements can be achieved than through redoubled, yet isolated, efforts. This movement is called Integrated Diagnostics and seeks to address maintenance and logistics support problems, beginning with the design phase of a new system. The objective is to increase the operational readiness of these systems to perform designated missions. More information on integrated diagnostics can be found in the proceedings of the Conference on Integrated Diagnostics (National Security Industrial Association, 1983).

Illustrative Scenarios

The two scenarios presented below dramatize the problems faced in maintenance today and how creative solutions might be implemented resulting in an improved maintenance situation tomorrow.

Petty Officer Today

The following scenario is extracted from Gross (1984) and illustrates, from a Navy perspective, the maintenance situation today.

> Let's imagine a technician sitting in the middle of the Indian Ocean, standing watch and operating the surface search radar which is one of the critical systems on a ship. He knows where he is and who else is around and doesn't want to run into other people. He may be on the night watch, and playing pinochle with a couple of buddies, and all of a sudden, about midnight or 1:00 a.m., every amber light on the power panel lights up and someone says, "Holy

cow, what's going on?" They've gone into a hard down situation. What does the technician do? Immediately, he picks up the maintenance manual, which is 9 inches thick and weighs about 15 pounds. He picks up about seven or eight pieces of general purpose electronic test equipment, walks over to the panel, starts playing with the built-in test, and goes through a routine of trying to fault isolate and detect what's wrong with that machine. Ultimately, if he's lucky, in a few minutes he fault isolates to an ambiguity group. If he's not lucky, sometimes it can be several hours. In the meantime, the CO of the ship is saying, "What the heck is wrong with my surface search radar? You took my eyes away." This poor technician is working with the tools we've given him which are, at best, barely adequate. If he's a smart tech or a super tech (and we do have some excellent technicians out there), he pulls out a little black book. If this thing has happened before, then he's got some information on it and he can go ahead and maybe solve the problem. If not, he's got a real problem. He's got to call the supply officer and say, "Hey, Mr. Porkchop, do you have seven or eight or nine modules," whatever the ambiguity group is. "I need to replace them." If he's lucky, he might have what's called a maintenance assist module. This allows him to take a "golden module" and start "easter egging" by random trial and error to get down to a faulty card, or reduce that fault group to a smaller number.

In our example we'll say he's lucky and they have all the spares on board to solve this specific problem. So he pulls the specific module out, replaces it, runs through an operational test, and he's back on-line.

Well, what happens to the module? Right now, if we're talking about the surface Navy, they go back to either a shipyard or a contractor. That can be disastrous in some instances. If that was the only spare on the ship, you probably won't see another spare back on the ship for 4 or 5 months. In the meantime, you're probably going to experience a failure. So, if you're lucky, you have some capability on the ship to repair these modules. They are sent to the technician who starts running them on the ATE. The technician runs through all eight or nine modules and says, "Hey, I got two bad modules." The other ones are put back into the supply system as ready for issue and two modules must be tested. What's happened here is that we've lost a bit of information. There's been information from interrogating and isolating those modules that we haven't transferred to that technician. This technician now runs into the same problem that the person taking it out of

> the prime system does. It gets put on a piece of automatic test equipment, gets run through a test program set and a diagnostic procedure, and lo and behold the technician gets it down to two, three, maybe four devices. Unless there is an intelligent probe or some other technique, we're dead in the water. So all the chips get replaced.

A comment about this scenario is in order. This scenario focuses on shortcomings due to or caused by maintenance philosophy, ATE, job performance aids, and logistics support. It assumes that Petty Officer Today is both competent (e.g., highly trained) and lucky; a situation that cannot be taken as the norm. Were Petty Officer Today less well trained, less able, or less lucky, the scenario could have been disastrous. With this in mind, solutions to the problems posed by this scenario include changes and improvements in maintenance philosophy, automatic test, job performance aids, logistics support, and training.

SSgt Bayshore

The following scenario is a vision of a solution. It presents a future maintenance operation and was adapted by Gunning (1984) from Johnson's Integrated Maintenance Information System: An Imaginary Preview (1981).

> SSgt Bayshore is now the crew chief for the new F-22 (Advanced Tactical Fighter). She begins her work day by reporting to the maintenance center and connecting her portable computer to one of the desk-top workstations. The day's work assignments appear on the screen. Aircraft 0808 has just returned from a mission and reports a radar system failure. The pilot debriefing report and BIT fault history, which were loaded into the system during the debriefing, are displayed on her screen. Bayshore studies the information and requests historical data for the radar unit and for aircraft 0808. As the data are retrieved, intelligent software in her system recognizes a pattern in the flight parameter data which matches a common radar system failure. The system recommends a course of action for fault isolation and lists the needed technical instructions. Bayshore indicates to Job Control that she is on her way to aircraft 0808. She disconnects the portable computer from the workstation and inserts a memory module which has been loaded with the needed technical orders, historical data, and diagnostic routines.
>
> She carries her 10-pound system to the flight line, opens one of the access panels, and plugs her portable computer into the technician's interface panel. (She remembers the story the old chief had told her years ago about how much time was wasted crawling inside the cockpit every time they had to work on an aircraft. But, that was when it

took more than a half-hour for aircraft turnaround.) She begins the fault isolation process by interrogating the avionics central computer. Her display draws a diagram of the current configuration of the self-repairing avionics network. She requests a comparison of the current configuration and the fully operational configuration. She notices that the radar and radar data bus interface have been operating in a re-routed configuration. Through the on-board panel, she initiates a system BIT test. The BIT report agrees with the pilot's debriefing, a radar malfunction has occurred. However, when the computer had analyzed the historical fault data, it discovered that 75 percent of the radar fault indications were caused by wiring problems and not faulty radar modules. She decides to test the wiring before removing a potentially good radar.

SSgt Bayshore activates the intelligent diagnostic aid which automatically downloads information about the current wiring configuration. Instructions appear on the screen showing her where to locate the wire bundles which might cause a radar fault indication. SSgt Bayshore unplugs the portable computer and walks to the indicated access panel. She opens the panel, locates the bundle, and begins the fault isolation process. The smart diagnostic software sequentially selects the optimum test point and displays graphic illustrations showing her how to conduct each test. (By now the system knows that SSgt Bayshore always requests graphic instructions and displays them automatically. When SSgt Bayshore was inexperienced, she had to select the graphic data each time, until the system "learned" what to expect.) After 10 minutes, Bayshore has isolated the problem. A bent pin on one of the connectors has caused the fault indication.

SSgt Bayshore calls up a diagram of the connector and indicates that she needs to order a replacement from supply. The information is automatically transmitted over the radio to the supply computer. The supply computer evaluates the requisition and responds by transmitting a status report. The part is in stock and will be brought to the aircraft in 10 minutes. Automatic monitoring programs update Job Control on the aircraft status and the availability of the required part. By the time Bayshore removes the bad connector, the van arrives with the replacement part. She replaces the unit and begins a final aircraft checkout.

She calls up a display of aircraft 0808's flight schedule for the next week. A heavy week of flying is ahead. SSgt

Bayshore asks for a comparison of the system capabilities needed for the upcoming missions and the capabilities of the current avionics configuration. She's in luck, the system has not degraded to a point where it needs repair. All critical systems are backed up with sufficient redundancy.

Now to check for projected system failures. She calls up the analysis of historical flight data which was performed back at the workstation. The analysis shows that an electrical system failure is likely to occur within the next 10 flying hours. SSgt Bayshore checks out the indicated subsystems and replaces the aircraft battery before finishing the checkout.

With her job finished, B~yshore returns to the maintenance center and plugs her portable computer into the workstation. She completes the needed maintenance reports for the morning's work by selecting a report option on her display. All the information, which was recorded as she worked, is automatically formatted and transmitted to Maintenance Analysis, to Job Control, and to the historical data base for aircraft 0808. SSgt Bayshore doesn't need to waste time filling out numerous reporting forms.

Bayshore again checks her work schedule for the day. No jobs for the next 2 hours. She decides to run through a training package for the new flight control system which will be installed next month. She relaxes in the maintenance center and plays with the graphic simulation model of the new system. She remembers how boring and difficult the classroom training was before the new training system was installed. Before signing off, she is reminded that she has only one more skills test to complete in the "MAZE" or maintenance activity simulation environment before she is eligible for promotion. She asks for an analysis of her training profile to determine weak areas, and then asks for her absolute and relative maintenance performance ratings. She is informed that she has had adequate simulated and actual practice in each area and that her fault detection and procedural tasks efficiency ratings have improved significantly. Furthermore, her standing in comparison to other E-5 crew chiefs is still within the promotion window.

SSgt Bayshore enjoys her work in the new maintenance operation. Now she is certain that she made the right decision when she left the airline to begin a career in Air Force maintenance.

The SSgt Bayshore scenario represents an ambitious set of goals for the maintenance community. The realization of these goals will depend largely on AI research.

Ways AI Can Help

First, what is AI? AI is the science and technology of reproducing human-level intellectual competence with machines. That is, AI is the practice of building process models of intellectual activity that can be run on a computer. The main intellectual activities of interest include problem solving, learning, and natural language processing. These activities generally involve complexity (designing a bridge), uncertainty (deciding whether to buy or sell on today's stock market), or ambiguity ("John said Jack said he went to the store."). All of these activities involve knowledge and the manipulation of knowledge in achieving a goal. Taking problem solving as an example, the basic AI approach is to create a space of all possible sequences of allowable problem-solving steps and then search this space for a sequence that leads to a valid solution. This search is neither random nor exhaustive, it is guided in order to limit the number of potential solutions considered. This example illustrates the two central issues of artificial intelligence: representation of knowledge and methods of controlling a search. In general the objective is to arrive at a good solution most of the time as opposed to the best solution all of the time.

How might AI help in solving modern maintenance problems? If we can get computer-based systems to do more of the human-level intellectual tasks required in maintenance, then AI will be of assistance. McGrath (1984) presents a good survey of how AI can help: (a) by reducing diagnostic errors through "smart" built-in test and knowledge-based expert systems for troubleshooting; (b) by enhancing maintenance training technology, for example, intelligent computer-assisted instruction and intelligent maintenance simulation; (c) by improving handling of technical information, especially in the areas of research and creation; and (d) by improving the testability and fault tolerance of systems through computer-aided design and engineering. More fundamentally, AI can help solve modern maintenance problems because it is interdisciplinary, sharing much of the two principal disciplines of psychology and computer science.

In the following section the principal subdisciplines of AI that have potential applications to maintenance are reviewed. A more thorough discussion of AI-related issues is presented in subsequent chapters.

Expert Systems

It seems appropriate to discuss the subdiscipline of expert systems first because it exemplifies many issues that span the field of AI. Although the first expert system was introduced nearly a decade ago, recent successes in domains like medical diagnosis, geological prospecting, and configuring large computer systems have attracted the attention and enthusiasm of military and industrial

personnel. In fact, most of the AI systems mentioned in the Proceedings of the Joint Services Workshop (AFHRL, 1984) are of the expert system type.

A number of interesting and difficult tasks require massive quantities of specialized knowledge that most people do not have. Programs whose performance is in this expert class are called expert systems and the construction of them is called knowledge engineering. Expert systems are suitable for a large number of diverse applications such as interpretation, diagnosis, planning, debugging, instruction, prediction, design, monitoring, repair, and control.

One of the guiding principles of expert systems is that problem-solving power lies more in knowledge than strategy. Thus, an important characteristic of expert systems is their reliance on large data bases of knowledge. Most expert systems use production rules to represent knowledge, since it is important to separate knowledge from the reasoning engine.

Though expert systems have demonstrated extraordinary performance in certain domains, they have a number of shortcomings which are listed below (Buchanan, 1982; Hart, 1980):

- inability to deal with problems for which their own knowledge is inapplicable or insufficient
- lack of ability to check their own conclusions
- narrow domains of expertise

We can expect many of these limitations to be mitigated as research and practice evolve more sophisticated expert systems in the near future.

Problem Solving

In AI, problem solving usually refers to the ability to solve nontrivial problems. Problem solving typically relies on heuristically guided search techniques which exploit domain-specific knowledge to prune search spaces. The object of these problem-solving procedures is to discover a path through a problem space starting at an initial situation and ending at a specified goal situation. This exploratory procedure can progress in either a forward or backward direction, depending on whether the search is data-directed or goal-directed. It employs a number of heuristic methods usually referred to as weak methods: generate-and-test, hill climbing, breadth-first search, best-first search, problem reduction, constraint satisfaction, and means-ends analysis.

Most major problem-solving systems combine one or more of the above strategies with some knowledge representation mechanism. They can also provide ways to divide the domain problem into smaller pieces, each of which may be solved more easily. Separate results are then recombined to form a single consistent solution to the original problem.

Problem solving is difficult to separate from knowledge representation because inferences can be made based only on what is known and on how that knowledge is structured. Problem solving uses knowledge representation as a framework within which to manipulate knowledge. General heuristics can be very powerful manipulators when applied in appropriate context, but it is an open issue as to how to construct a representation that forms a basis for heuristics. The relevance of problem solving to the maintenance task is ubiquitous, since nearly all aspects of maintenance can easily benefit by employing powerful problem solvers.

Planning

Planning refers to the process of computing several steps of a problem-solving procedure before actually executing any of those steps. In fact, planning is a very close relative of problem solving. Planning usually involves methods of decomposing large problems into manageable subparts, focusing on ways of handling and recording interactions among the subparts as they are detected during the problem-solving process.

Problem solving would almost always be successful if the world provided perfect information. However, since there is gross randomness in the world, special difficulties come up in deciding sequences of actions. The question that arises is, must we completely abandon a present strategy in order to replan, or should we attempt to maintain some kind of problem metastructure and just patch the procedure when required by circumstance? The relevance of planning to the maintenance task is most apparent in recovery and compensation for system failure.

Natural Language Understanding

Natural language understanding is a translation process, requiring a mapping from text, dialog, or some other language representation into a second representation. The second representation is usually chosen to correspond to a set of actions to be performed as a result of an appropriate translation.

Natural language understanding should be distinguished from natural language interfaces where the target representation is ordinarily a sequence of commands. Programs which map English to a set of 10 actions, for example, are better off not enduring the troubles and complexities of natural language. It is simpler just to instruct users to press buttons or issue coded commands.

Indeed, programs should be capable of being told what to do, but if they are unable to solve a large number of problems by taking advantage of the richness of natural language, they become impractical. Again note that one of the important underlying issues is representation, in this case the target representation. Natural language finds relevance to the maintenance task in many ways, from training applications to document understanding and production, and especially in data base query systems.

Learning

Learning is usually taken to mean the ability to adapt to new surroundings and to solve new problems. Two important components of learning are the acquisition of new knowledge and the problem solving required to integrate new knowledge (a mapping problem) to deduce new facts from incomplete information.

One of the problems encountered in discussing learning systems is the matter of definition. What exactly is meant by learning? Machine learning systems can be broken down in one of two ways: on the basis of underlying strategies where the processes are ordered by the amount of inference performed by the system (e.g., learning by rote, by analogy, from instruction, from examples, or from observation and discussion) and on the basis of representation of knowledge or the type of knowledge acquired (e.g., through parameter adjustment, decision trees, formal grammars, production rules, formal logic, graphs and networks, or frames and schemata).

Learning is similar to other kinds of problem solving in that it requires an organized store of information (representation), the ability to generalize from particulars, and the ability to focus on a promising direction. For this reason, learning programs confront the same difficulties as other problem-solving programs. One major issue is the credit assignment problem, the matter of assigning responsibility to individual decisions that led to some overall result. Another critical issue is the choice of the correct set of primitives for representing requisite knowledge. Learning is particularly relevant to the maintenance task for trend analysis, signature extraction, and model building.

AI Systems Engineering Issues

Hardware/Software Issues

A number of hardware and software issues are also relevant in applying AI to maintenance and troubleshooting. Important considerations include the following questions: Are there hardware/software packages readily available to facilitate the development of such systems? Is LISP a necessary ingredient? What are the computing resources required to support these systems? Are such systems viable in real-time response environments? What kinds of user-interface technologies are available? Clearly, the answers to these kinds of questions will be different depending on the kind of system being built. Also, since the entire area of hardware and software is rapidly developing, this summary attempts to describe where things stand now and where they appear to be heading.

Basic support. Providing basic hardware and software support is a principal concern of all AI projects. Historically, most AI work tends to be LISP-based. Early AI systems were developed primarily in INTERLISP and MACLISP on DEC-10 systems running noncommercial operating systems. This made portability and accessibility a serious problem for projects outside the main

AI research labs. This situation, however, is changing for the better in several ways. Versions of INTERLISP are now available for VAX machines running UNIX or VMS and on Xerox's 1100 series of personal work stations (Dolphins, Dandelions, Dorados, etc.). This has prompted experiments in "porting" a variety of AI tools (such as EMYCIN and KL-ONE), which somewhat improves their availability. INTERLISP itself requires significant amounts of computing resources, and current experience with both the VAX UNIX and Dolphin implementations suggests there are still serious performance problems to be overcome.

An alternative is to use LISP machine hardware available from SYMBOLICS or LMI (both of which are independent MIT spinoff companies). They provide MACLISP-based systems which can be configured to provide significant computing power. The primary difficulty with such systems is justifying their cost as a one-person work station since they cannot be time-shared for more than one application or for multi-person development. However, multi-user stations are beginning to become available.

Another hardware and software problem is the proliferation of LISP dialects. For example, INTERLISP-based software can be very difficult to import into a MACLISP environment. Consequently, the fact that some dialect of LISP is available on a particular machine does not guarantee immediate access to the large body of AI tools written in various dialects of LISP. Nor does it guarantee that systems developed locally will be easily moved to other machines. There is currently an attempt to define a Common LISP language to improve portability problems. However, it will be several years before such standardization will have any effect.

Franz LISP is an interesting alternative. Developed by the University of California at Berkeley, Franz LISP is a dialect which runs on both UNIX and VMS VAX systems. Several MACLISP-based systems as well as OPS5 have been ported into Franz LISP with only minor conversion problems.

There are several features of Franz LISP which are useful. First, it admits to the existence of other languages, providing mechanisms for calling routines written in other high level languages, such as C and FORTRAN. Second, there is a fairly high degree of symmetry between compiled and interpreted code allowing one to easily intermix the two and incrementally improve performance as routines stabilize. Finally, there is a growing number of conventional microprocessor-based systems which support Berkeley UNIX and for which Franz LISP is available. Thus, a number of low-cost alternatives to dedicated LISP machines and/or time-shared minis or mainframes are available, for example, the SUN work station, which is Motorola 68000 based and runs the same Berkeley UNIX and Franz LISP as the VAX 780. This, however, is only one example, for there are more machines being announced all the time.

One important advantage of the various "personal work stations" currently available is the quality of the user interface. All come with high-resolution black/white bit-mapped displays (color is optional), a mouse input device, and software to support "windowing" and graphics. These features can significantly improve the development process as well as the ultimate user

interface of an expert system and are difficult to duplicate on more conventional systems.

In summary, there is a variety of hardware currently availabl: to support the application of AI to maintenance and troubleshooting. The difficulties arise in trying to provide a reasonably uniform software base on which to do the development work. Even if one insists on doing everything in L'SP, dialect dependencies get in the way. There are some recent developments which have the potential for reducing such problems but they are too premature to evaluate at this time.

Faced with the above concerns, one may legitimately ask, "Why LISP?" Is this just the tyranny of tradition or is there something about LISP that allows things that cannot be done in FORTRAN or PASCAL? The answer appears to be both "yes" and "no." LISP and its associated programming environment allows rapid prototyping of complex systems in a way that most traditional programming languages do not. This is particularly useful when the only way to assess the merits of alternative designs is to implement them, subsequently abandoning one or both. A second argument (weakened by dialect dependencies) is that there is a large body of useful software already written in LISP that one wants to exploit rather than rewrite. This is, of course, the same argument that has kept FORTRAN and COBOL around long after what some consider their intellectual demise. A third argument is that experienced AI practitioners are accustomed to LISP and therefore it would be difficult to recruit AI talent if LISP were not the language of choice in the prospective environment.

Higher level software issues. Ideally, the system builder will select an appropriate set of AI tools for the particular system that is to be fabricated. Unfortunately, the availability of such tools is currently a serious constraint. At the highest level, there are mature expert systems for particular problems, such as DENDRAL (Buchanan & Feigenbaum, 1978), MYCIN (Shortliffe, 1976), PROSPECTOR (Hart & Duda, 1977), and CADUCEUS (Pople, 1982). These systems could theoretically be an ideal starting point for new applications with similar characteristics. In practice, however, they may be available only within research projects, currently unsupported, proprietary, or lacking user and system documentation. It can turn out to be far simpler to reimplement basic concepts locally than to move an implementation to a compatible and accessible machine.

Even if such moves could be made with relative ease, the lack of domain-independence of the implementation is a serious problem. Several attempts have been made to alleviate such difficulties by extracting the "essence" of a particular system, for example, EMYCIN (van Melle, 1982) and HEARSAY III (Erman, London, & Fickas, 1981). The essence of a system is its knowledge representation and problem-solving mechanisms, leaving application-dependent knowledge to be filled in by the system designer. Currently the availability of these derivative systems is not much better than the availability of the original ones, although this is slowly changing with the emergence of several knowledge engineering companies attempting to provide commercial-grade software.

There are a number of other domain-independent tools. Examples of such tools, but certainly not an exhaustive list, includes OPS5 (Forgy, 1981), KRL (Bobrow & Winograd, 1977), KMS (Reggia & Perricone, 1981), ARBY (McDermott & Brooks, 1982), ROSIE (Fain, Gorlin, Hayes-Roth, Rosenschein, Sowizral, & Waterman, 1981), and PROLOG (Clocksin & Lellish, 1981). Such tools attempt to provide support for one or more of the basic components of an expert system (typically, the knowledge representation and a basic inference mechanism). Again, the problem of availability is being resolved slowly and the usefulness of these tools will be decided by users when they are more accessible. It should be noted, however, that commercially developed building tools are appearing in the literature almost daily.

One point that should be clearly made here is the difference in the kinds of tools required by the system designer. While autonomous, consultant, and training systems require one or more underlying knowledge representations and inference mechanisms, the role of the human-machine interface plays an increasingly important role as the move is made from autonomous to consultant to training systems. It is fair to say that most of the AI tools developed to date have focused more on knowledge representation and inference than on interface issues.

In summary, the system builder currently has a variety of powerful conceptual AI tools for representing and reasoning about knowledge which, in the near term, will require local reimplementation.

Knowledge Acquisition

The system designer, even after choosing a knowledge representation and inference mechanism, still faces the difficult task of effectively capturing the domain knowledge required to provide the desired level of performance. A common approach is to find a domain expert willing to submit to endless hours of conversation, interrogation, and argument in an attempt to discover how that individual solves problems. Typically, this expert also serves as the end user of the developing system to provide feedback on its performance. This task of extracting and coding the requisite knowledge, "knowledge engineering," cannot be underestimated. It is a serious commitment on someone's part to spend what could be several years immersed in the intimate details of the application area.

There have been several attempts at minimizing the role of the knowledge engineer in the acquisition process by providing a set of tools that can be used by the domain expert to build, debug, and extend the knowledge base (e.g., Davis, 1976; Reboh, 1981). Such techniques have enjoyed only limited success and have been tightly bound to a particular system. They should be viewed as exploratory in nature.

Also exploratory but showing considerable promise are the attempts to automate the knowledge acquisition process via forms of machine learning. For examples, see Michalski (1980) and Holland (1980).

Even if a knowledge base is reasonably well developed, there are still significant questions about verifying its correctness and completeness and maintaining consistency over time and across several experts. To summarize, knowledge acquisition in the near future will be achieved with considerable investment of time and effort.

The User Interface

Most people have been exposed to more than one software system where difficulty of use or intolerable response time clouds any appreciation of the system's technical merits. These problems have increasing significance for AI technology when it is moved out of the laboratory and into the applications world where a decision aid may be accepted or rejected primarily on the basis of the quality of the user interface. In this area, as in knowledge acquisition, there are some technological developments emerging to help system designers. As mentioned above, the introduction of a variety of personal, LISP-based work stations with high-quality displays. mouse input devices, and software to support windowing and graphics has provided a significant improvement over the more common CRT interface. Also available, but not extensively explored, are technologies such as touch-sensitive screens, joy sticks, and videodiscs.

A good deal of independent work beginning to affect the expert systems area is in natural languages. Current systems, however, tend to have languages that are highly stylized to a particular application and, at best, embody only limited forms of "natural" language.

Improvements in response time are presently achieved by employing faster hardware or by introducing domain-dependent heuristics into system components initially designed for generality and domain independence. There are several other alternatives being explored. One approach involves the development of "compilation" procedures for converting knowledge bases from a high-level form (useful while building and debugging) to an efficient low-level representation for use by the end user. Another approach is to develop and exploit parallel architectures, such as ZMOB (Rieger et al., 1980) or NETL (Fahlman, 1979), for use in expert system design. Both of these approaches are still experimental at this time.

Summary

In this review of the hardware and software issues, there is concern that the reader may come away with negative impressions of the state of the technology. That is certainly not the intent of this section. The standards which have been set for AI systems and the techniques used to build them are very high. There have been noteworthy achievements (such as DENDRAL, CADUCEUS, and PROSPECTOR) and more can be expected. However, it is important to understand the commitment (in terms of hardware, software, and people) that is currently required to build a system of "one's own." It has been indicated in the field that the knowledge engineering business is now a "cut and dried," 2 to 3

month process using off-the-shelf hardware and software packages. Observation and experience suggest that this is the exception and not the rule.

Pragmatics of AI Research

Despite the current favorable research climate, AI remains a costly, time-consuming, and risky proposition. Practical considerations often dictate whether or not a proposed AI project is successful in attracting support and producing a worthwhile product. Participants in the Joint Services Workshop (AFHRL, 1984), especially those involved with program management, suggested two complementary strategies for AI practitioners: target high payoff areas and minimize risk factors.

Target High Payoff Areas

The potential payoffs from successful AI applications in maintenance are enormous and it is this potential that has drawn so much attention at the program level (Shumaker, 1984). However, the payoffs cannot be taken for granted. Perhaps the best advice is to be responsive to the user's needs. One way to do this is to select existing equipment for the research test bed rather than equipment that is still under development. Although this may cause additional problems because the proposed project must be retrofit, the benefits to maintenance are easily demonstrated. Similarly, basic research should be designed so that the results are easily transitioned to real-world maintenance applications.

Another way to maximize the payoffs from AI is to focus on problems that generalize to a broader range of hardware than the specific research test bed. At the very least, test bed hardware should be representative of a larger family or class of equipment. A much higher payoff would result from the development of generic AI products (e.g., system building tools) that are directly applicable throughout or even across equipment domains.

Minimize Risk Factors

The risks associated with AI research can be minimized in a variety of ways. First, the researcher can adopt a fairly conservative approach that limits the scope of the problem under investigation, exploits existing technology, ensures a stable research environment, and focuses on well defined and understood problem domains (e.g., electronics).

Second, researchers should carefully consider the availability of project resources and the state of current technology to support their work. Experienced AI practitioners are a scarce commodity. Technological constraints can also be important, particularly if the long-range research plan calls for scaling up a demonstration project to deal with more complex real-world applications.

Third, risks can be minimized by planning a modular project structure within a reasonable time frame. An AI project can require 10-15 years to complete, but few program managers can wait that long for results. A modular approach provides intermediate milestones that help maintain high levels of interest and visibility throughout the course of a lengthy project (Shumaker, 1984). Further, even if the overall project goal is not realized, there can be positive results and tangible spin-offs from the effort.

Finally, in order to be successful, AI projects should actively promote user acceptance at a variety of levels. This means building systems that not only have a user-friendly interface, but are able to adapt to the needs of individuals. As Coppola (1984) notes,"AI systems, must, to the extent possible, be designed so that the human will consider it as a partner rather than as an inanimate tyrant . . ."

Final Caveats

Nearly all of the AI workshop participants had words of caution for their colleagues. The maintenance problems facing the services are real and difficult, and while the promises of AI are great, it is not the panacea people sometimes suggest. Even if the advice presented above is followed, there is no guarantee of success. There are serious pitfalls, such as natural language interfacing, that should not be underestimated. Overall, the climate for AI research seems to be one of guarded optimism: rapid advancements are being made, but expectations must be kept in check.

III. AUTOMATED SYSTEMS FOR MANAGING HARDWARE FAILURES

The failure cycle has three major components: detection of system failure, diagnosis of the failure, and recovery from the failure. There is a substantial amount known about fault detection. This fact is suggested by the extensive knowledge of fault mechanisms. Techniques drawn from fault-tolerant computing and concepts from BIT can be employed in fault detection. Hence, fault detection can be significantly automated. However, much less is known about diagnosis, even when done by human technicians. Some diagnostic processes are now yielding to automation, as evidenced primarily by ATE and early results in AI. Recovery techniques range from real-time work-arounds to physical replacement of hardware.

The goal of this chapter is to suggest ways in which AI can aid the maintenance process at various points in the failure cycle. Each of the following sections on detection, diagnosis, and recovery will describe machine approaches and discuss applicable AI methodology. In conclusion, there is a short discussion of fault prediction, the brevity of which is dictated by the paucity of knowledge in this area.

Detection of System Failure

Detection is the process of a human operator or automated equipment determining that a failure event has occurred; i.e., that a circuit is not operating correctly. To decrease the possibility of failures, various fault-avoidance techniques may be employed. Examples are environmental modification, use of high-quality components, and use of high levels of component integration.

Machine Fault Detection

Fault detection deals with the inevitability of failure. In hardware, fault detection techniques supply warnings of faulty results. They may also provide limited diagnostic capabilities, resolving to a finite number of possible failure locations, such as a device or an ambiguity group of devices. The key to fault detection is providing extra information or resources beyond those needed during normal system operation. This added information is not used to detect failures, but to detect the faults and errors that are caused by failures. Action following detection can range from ignoring the failure to retries or even automatically switching in new components. Retries are often successful with transient or intermittent faults. Four important hardware methods of fault detection are duplication, error detection codes, watchdog timers, and consistency and capability checks. None of these fault detection methods escapes the classic dilemma of "Who checks the checker?" problem can be mitigated with additional cost, complexity, or performance degradation, but it cannot be completely resolved.

Fault detection for electronic devices is usually accomplished with hardware, in which case either some kind of visual or auditory warning is invoked, or it is accomplished by the human user through pattern-based recognition.

Techniques like BIT are certainly useful and fairly successful as far as they go. The problem with BIT, however, is that it is based on failure modes predicted by designers from design specifications. This means that BIT will work only for a set of preconceived failures, possibly omitting some failure modes due to design oversight. Furthermore, BIT itself relies on hardware or software algorithms constructed by fallible humans who may overlook important parameters or conditions.

It has been suggested that "smart-BIT" could overcome many of these shortcomings in automatic failure detection. While this is doubtless true, there may be some argument about exactly what constitutes smart-BIT. The four methods of hardware detection listed above are "smart" methods, but people practiced in fault-tolerant hardware design would not label these methods smart, much less AI. The ability to do thresholding or voting certainly increases automatic failure detection capabilities, but these can be easily implemented in hardware with no appeal to AI.

The key to both detection and diagnosis of failures is information. Most systems are designed such that a failure cannot be detected until it has perturbed the system at a fairly high level of abstraction, despite possible early manifestations of failure at significantly lower levels. Fault-tolerant hardware or error-correcting hardware often masks such errors, preventing them from corrupting operational processes. If this low-level information were observable, it would be extremely useful for fault detection, since many devices fail soft before failing hard.

This concept of internal observability and controllability is discussed by Grason and Nagle (1980). Techniques of design for "testability," some of which require additional hardware and others which do not, include: avoiding one-shots when possible and if not possible, controlling and observing their outputs with test points; partitioning the circuit into functionally independent subcircuits for testing and placing test points between subcircuits; breaking reconvergent fan-out paths when they interfere with testability; using elements in the same integrated circuit package when designing a series of inverters; and trying to assign gates in a feedback loop to the same integrated circuit package.

AI Applications to Fault Detection

There are practically no current applications of artificial intelligence to fault detection. However, there are two areas that are ripe for application of AI techniques: trend analysis and automated design aids.

Trend analysis. Trend analysis is suggested by the fact that before a piece of equipment fails, it undergoes a period of increasingly unreliable behavior. In other words, most hard failures are preceded by a period of intermittent

failures. Often diagnostic programs cannot recreate an error event because they do not stress the system in the same way that operational software does. By designing for testability, performance information from low levels could be collected by an error logging program. Error logging captures information about the state of the system at the time of the error, thus providing clues to the source of the error. A program could periodically scan the log searching for patterns and trends. Some of the AI issues involved are:

- automatic characterization of normal system behavior (normal conditions may differ, even across different instances of the same system)
- automatic extraction of patterns or signatures
- automatic selection of tests based on observed signatures

Automated design aids. A hardware designer could be significantly helped by automated design aids when designing for testability and maintainability. There are already silicon compilers to assist in VLSI (Very Large Scale Integration) design and large data bases of preconfigured chip layouts. A prototype expert system for automating design for testability would function as a "testability" expert or designer's assistance, checking for design-for-test rule violations. If a violation is found, the system automatically transforms the design to remove it.

Diagnosis of System Failure

The diagnostic task consists of five steps which, when repeated iteratively, converge on a fault:

1. Decide whether further diagnostic refinement is warranted.
2. Select where to measure next, such that expected information gain per unit cost is maximized.
3. Identify the expected value of the selected measurement.
4. Make the measurement.
5. Determine the implications of this measurement in terms of component blame or innocence.

This process may be summarized as a cycle of making measurements and computing entailments.

Fault diagnosis is a special kind of problem solving, sometimes called classification problem solving (Clancey, 1984), in which the problem solver selects from a set of pre-enumerated solutions. Diagnostic test strategies may be precomputed, as in the traditional ATE approach to diagnostic test, or they may be developed in real time as the diagnostic session proceeds, as is typical in an AI approach. In either case, the set of "right answers" (e.g., potential faults) that a successful strategy converges toward is known in advance.

Approaches to system diagnosis fall into two distinct categories: symptom-based and specification-based. These two approaches are evident in human and machine-based diagnosis. The symptom-based approach, often termed shallow reasoning (also termed evidential, associationistic, or empirical reasoning), solves diagnostic problems by manipulating a set of associations between symptoms and faults. With this approach, the associations between symptoms and faults are heuristic in nature (e.g., not infallible) and based more on experience than on reasoned causal derivation.

The symptom-based approach to diagnosis may also employ tactics for capturing the times and locations of observed errors. This aspect is appealing because it bears so much similarity to what a technician might observe in a failing system, though automated systems possess greater capabilities than humans. Normal system behavior is contrasted with error behavior, usually by discovering or analyzing trends in data. This has the advantage that many intermittents can be successfully dealt with and that certain classes of failures can be predicted. Work is still in progress on this approach, but the early returns are promising. An example of this approach is a rule-based system to be incorporated in the B-1 aircraft to monitor and analyze BIT and sensor parametric data in-flight.

In contrast, the specification-based approach, often termed deep reasoning (also termed causal or state-based reasoning), solves diagnostic problems by reasoning from the structure and behavior of the device. The structure is a description of the connectivity or dependency of its components. The behavior is a description of the input-output behavior of each component. Using these descriptions only, the composite behavior of the system can be derived through the propagation of individual component behavior through the connectivity network. This propagation is constrained by applicable network laws, such as Ohm's and Kirchoff's Laws. Often multiple possible composite behaviors are generated through this causal propagation. Knowledge of the device's intended purpose or function can be used to rule out incorrect derivations of composite behavior (de Kleer, 1979).

If the diagnostic program is being developed directly by a test engineer, then the qualitative causal model of the system under test is in the engineer's mind. If the diagnostic program is AI-based, then the model is in a computer. In either case, this model is used to generate expectations about circuit measurements which are compared with actual measurements. Discrepancies between expected and observed values are then incorporated in the model to rule out certain components and cast additional suspicion on others. As described above in the basic diagnostic cycle, based on the new state of the model, a new measurement is selected that would yield maximum information gain.

AI has been developed for both symptom- and specification-based techniques. Human technicians also use either, preferring pattern matching whenever possible and resorting to deep reasoning only when forced to do so.

Machine Fault Diagnosis

The practice of diagnostic test program set development is continually evolving. The direction of this evolution is toward increasing use of computer-based aids in automatic test program generation (ATPG). The use of such aids is developed more for digital circuitry than for analog or hybrid circuitry.

For digital circuitry, digital ATPG is an engineering reality. A model of the unit under test (UUT) is developed from numerous sources of information including schematics, parts lists, test specifications, a model library of digital circuit components, and lists of input and output pins. Then an ATPG facility such as LASAR or HITS is used to generate stimulus patterns, simulate unfaulted circuit behavior, and with selected faults, simulate faulted circuit behavior. This latter phase yields statistics and useful information such as percent fault detection, lists of undetected faults, or a fault dictionary. Postprocessing in the ATPG facility yields the test program set (TPS) in an ATE programming language such as ATLAS or JOVIAL. The TPS and necessary interface adapters for correcting the UUT to the ATE then undergo engineering evaluation and system compatibility tests. End products are deliverable documentation and the TPS in a digital working media.

For analog and hybrid circuitry, less automation is available. Test program sets are developed by a test engineer working from a variety of sources, including test requirements, drawings and schematics, field maintenance data, reliability and maintainability handbooks, and old TPS. The remainder of the process (test, evaluation, and documentation) is the same as for digital systems.

Ideally, ATPG should be conducted in parallel with the design process. During the design process, test sets should be built up in parallel with a testable circuit. In this way, the two processes would interact, converging on a testable design for which a test program can be reasonably generated. Test patterns would be generated for elementary circuit modules and then assembled into a complete diagnostic program. A combined automated design aid and automated test program generator would help the designer in appraising the diagnosability of the device or system under design and suggest modifications compatible with its ability to generate tests.

Four principal problems with the traditional diagnostic programs are:

1. Each diagnostic program must be created anew for each new device, even if the device is similar to another device for which a diagnostic program has already been written.

2. The coverage and accuracy of traditional diagnostic programs depends not only on valid fault models, but also on the experience, competence, and specification interpretation *skills of the programmer.*

3. Traditional diagnostic programs are almost always inadequate for locating transient and intermittent failures.

4. *System diagnostics* usually begin at a low level and test the entire system, often running for several hours before locating a problem. The constituent tests are not easily decomposed for assessing specific problems.

Solutions to these problems can significantly benefit from the technology *of artificial intelligence.*

AI Applications to Fault Diagnosis

The AI approaches to diagnostic problem solving, active for the past decade, have *shifted* the focus of attention from test generation to diagnosis. Test generation tells how, given a fault, to determine a set of input and output values which will manifest the fault. This strategy is most appropriate to exhaustive testing for equipment check-out or certification. Diagnosis, however, presents the task of reasoning from observed circuit misbehavior back to the responsible fault. The basic task is repair, not initial testing (Davis, 1982).

Focusing on AI approaches to diagnosis, three separate areas in the literature are of interest: (a) logic modeling, (b) specification-based approaches to diagnosis, and (c) symptom-based approaches to diagnosis. In addition, the literature on hierarchical problem solving is applicable to all of the above approaches.

Logic modeling. Logic modeling as a mathematical concept is treated in a series of papers by Wong and Andre (1976, 1981) and Andre and Wong (1975). Other treatments of logic include Longendorfer (1981) and Cramer et al. (1982). Severai proprietary applications of this technique to electronics diagnosis include LOGMOD (DETEX Systems, Inc., n.d.), STAMP (Simpson & Balaban, 1982; Simpson & Agre, 1983), and the FIND system, developed by the Hughes Aircraft Company.

These systems implement the structure modeling aspect of a specification-based approach to diagnosis; generally, they fall short of modeling behavior and purpose. Even so, such dependency models alone provide significant diagnostic leverage, either as a tool for the test engineer or as a diagnostic system per se. For example, these programs are very good at finding the best place to conduct the test such that the set of possible faults is split in half, something a human contemplating a large circuit schematic is demonstrably poor at doing. The main disadvantage is that while this approach provides information regarding where to test, it provides no information regarding the expected values, which must be computed by test engineers.

A sophisticated AI system based on logic modeling principles is INATE (Cantone, 1984; Cantone, Pipitone, Lander, & Marrone, 1983). Recently, this system has been extended to incorporate functionality as well as topology (Cantone, Lander, Marrone, & Gaynor, 1984).

The specification-based approach. Most work in the field of AI applications to electronics troubleshooting has focused on deriving diagnostic strategies from descriptions of device behavior, structure, and intended purpose. King (1982) has reviewed this literature, concluding that AI methods applied to troubleshooting devices can be regarded as "flow processing" systems, whose interesting properties arise from the behavior of and relationship between components. Logic modeling is subsumed by these methods, as the means of describing the connectivity of components. But the specification-based approaches also rely on a complete behavioral model of the modules in the system and a diagnostic strategy based on discrepancies between predicted and observed behavior of the system. The modeling focus is on correct, unfaulted performance alone; models of faulted performance are not needed.

As reviewed by King, work in this area began with LOCAL (Brown & Sussman, 1974), EL (Stallman & Sussman, 1977), DESI (McDermott, 1976), and WATSON (Brown, 1977). Additional work in model-based diagnosis includes SOPHIE (Brown, Burton, & de Kleer, 1982), DART (Genesereth, 1982), recent work at the Massachusetts Institute of Technology (Davis, 1983; Davis, Shrobe, Hamscher, Wieckert, Shirley, & Polit, 1982; Hamscher & Davis, 1984), and most recently, work by Pipitone (1984) at the Navy Center for Applied Research in Artificial Intelligence.

Specification-based diagnosis is but one task studied by researchers in the AI field of qualitative reasoning about physical systems. Other tasks that this research deals with include simulation, envisionment, mental models, verification, and deducing functionality. A recent volume of the journal, Artificial Intelligence (Bobrow & Hayes, 1984), is devoted to this subject, bringing together research previously published in scattered conference proceedings. Work not represented in this volume includes Moorthy and Chandrasekaran (1983), and Sembugamoorthy and Chandrasekaran (in press).

The symptom-based approach. In contrast to model-based approaches to diagnosis is the use of evidential rules to heuristically determine probable causes of failure based on observable symptoms. This is the most well-developed approach to expert systems in general, exemplified by the MYCIN system (Shortliffe, 1976). No causal model need be explicitly present in the expert system knowledge base for this approach to function. Indeed, this approach is most useful in situations where detailed and explicit causal models are often lacking or incomplete, such as in medical diagnosis. The symptom-based approach can also be used when a model does in fact exist, as is the case in electronics, but where the implications of the model are derived in the "mind's eye" of a knowledge engineer and entered into the expert system in the compiled form of symptom/fault associations. This approach to expert systems development is more tractable than the model-based approach, as evidenced by numerous recently developed systems designed to be usef[illegible] ndustrial tools. Systems include

ARBY (McDermott & Brooks, 1982), the Intelligent Maintenance Aid (Hinchman & Morgan, 1984; Williams & Hinchman, 1983), DELTA (Bonissone & Johnson, 1984), a related DELTA application at the Wright Aeronautical Laboratories (Davison, 1984) and LES (Laffey, Perkins, & Nguyen, 1984).

Hierarchical decomposition. Hypothesis refinement (also termed establish-refine) is key to efficient diagnostic reasoning. A fault is isolated to one of a set of probable causes at a given level of abstraction ("established"). Then the probable cause is broken down into more finely detailed probable causes ("refined"). The process is repeated until the fault is isolated within a sufficiently small probable cause (Chandrasekaran, 1983; Tanner & Bylander, 1984). This strategy is manifest in the three-level military systems maintenance philosophy of field, intermediate, and depot maintenance. However, even within a given maintenance level, this strategy of "divide and conquer" can yield diagnostic power and efficiency.

Integrated approach. As has been mentioned, human technicians prefer to employ a symptom-based approach, yet can resort to a specification-based approach if forced to do so. AI systems can employ a similar strategy, but to date they do not, tending instead to be either symptom-based or specification-based, but not hybrid combinations.

The two approaches are, however, inherently interrelated. For example, there must be a causal explanation for every empirical fact. The specification-based approach focuses on the causal explanation, the symptom-based on the known fact. With one exception, engineered systems capitalizing on the potential synergism between the two approaches do not exist. Fink, Lusth, and Duran (1984) describe an early implementation of a hybrid system, the development of which is a desirable goal for several reasons. First, the symptom-based approach suffers from the "knowledge engineering bottleneck" (Davis, 1982). Building empirical rule bases by hand is prohibitively labor intensive. Symptom-based systems will suffer from poor generality (the transferability of a rule base from one system to another system), poor robustness (the ability to deal with previously unencountered circuits or faults), and poor constructibility (the amount of human labor involved in developing the rule base). Alternatively, specification-based systems hold promise for highly favorable ratings with respect to these criteria. For example, one specific advantage is the apparent possibility of deriving the dependency model of system structure automatically from CAD/CAM engineering data. However, the specification-based approach is currently less feasible for near-term demonstration and application.

Recovery From System Failure

To recover from failure means that either the fault has been repaired or adequate compensation has been made. Repair suggests that the actual fault has been diagnosed and the fault component replaced. On the other hand, compensation suggests that the effect of a fault symptom has been mitigated and that functionality has been at least partially restored.

Machine Recovery From Failure

Machine recovery from failure is quite limited. The typical example is the Tandem NonStop system which automatically switches redundant components on- and off-line as necessary. Other common techniques are mapping out bad disk pages and reconfiguration of memories. Note that these are all compensation, not repair or replacement actions. There is some experience in self-repairing circuits, especially in VSLI where spare (not redundant) circuits are included on the chip and are utilized when primary circuits go awry.

One problem of automatic recovery mechanisms is that they have no knowledge of functionality, that is, no way to reason about a particular configuration of resources that permits selected priority functions to continue at the expense of others.

AI Applications to Recovery

There seems little need for the application of AI to recovery in the case of repair except in situations where a choice must be made among several possible repair actions. However, AI approaches to recovery in the case of compensation for system failure are interesting because to make appropriate compensatory actions, the following are needed:

- a model of the function of the system
- an understanding of what hardware provides what function
- a scheme for ordering the importance of various functions
- ability to plan sequences of actions
- knowledge of whether any compensating strategy would provide adequate recovery

The relevant AI issues in this case include representation (for modeling functionality and mapping function to hardware or vice versa), planning (a special case of problem solving), and metaknowledge or metacognition (knowing that you do not know something).

Fault Prediction

There are no known machine-based predictors of system failure. There are, however, examples of programs that attempt to predict such phenomena as weather and earthquakes. Such programs rely heavily on underlying causal mechanisms and models describing the ways in which they can interact. A causal model of the process under consideration is required in order to understand the relationships among actions, outcomes, and predictions.

Causal models are developed through the use of diagnostic inference. Past observations, events, and data are used as evidence to infer the process(es) of the past. People continually engage in shifting between forward and backward inference in both making and evaluating predictions in a manner analogous to shifting between top-down and bottom-up strategies in problem solving. An important consideration is how to determine when to make this shift.

Another important aspect of diagnostic inference as part of a prediction strategy concerns the process by which relevant variables are found and hypotheses formed. How do people distinguish between parameters relevant to a situation and those of lesser importance? One of the most critical aspects of prediction is to choose relevant cues to causality. There are four important points about cues:

1. The relation between a cue and a cause is probabilistic.

2. People learn to make use of multiple cues in order to mitigate errors due to overreliance on single cues.

3. Redundancy in the environment facilitates the use of multiple cues.

4. Multiple cues do not eliminate uncertainty, but they do reduce it.

It is possible to utilize certain common sense heuristics in evaluating the above points. Among them are temporal order of cues, the degree to which two variables occur together, contiguity in time and space, and the number of competing or alternative variables that appear to explain the same symptoms. Similarity plays a role in finding relevant cues, and the degree to which one variable can predict another is an important causal cue.

Aside from these crucial issues, any attempt at automated fault prediction will have to deal with the problems of gathering the right information and storing it in a compact, information-preserving form for later examination, since it is almost impossible to tell a priori what data will be critical for suggesting causation and what will not. Additionally, any useful prediction models will doubtless have to be constructed automatically based on observed events. This puts very strong requirements on learning by machine.

IV. DEVELOPING AND USING HUMAN RESOURCES

In Chapter II the difficulties of current maintenance systems and the limitations of BIT and ATE technology are summarized. BIT and ATE technology is an attempt to lessen the degree of dependence on human diagnostic skills because of human limitations as diagnosticians as discussed by Rouse (1984) and problems with personnel and training as discussed by Halff (1984). However, there are a number of problems associated with current BIT and ATE systems. For one thing they cannot handle all diagnostic tasks and, therefore, human involvement in diagnosis is still required. Yet these systems have poor or nonexistent human interfaces and do not exploit human problem-solving capabilities except as sensors and manipulators. They provide either too little information, a go/no-go signal, or far too much information, a string of hexadecimal digits on a small CRT in a cockpit. Thus, it is the failure to exploit human problem-solving capabilities and poor system-human interface, that combine with human limitations and training problems to exacerbate an already difficult situation by increasing the complexity and costs of maintaining state-of-the-art systems.

The primary assumptions for this chapter are that it is possible to build more effective and less costly automated diagnostic systems if these systems exploit human problem-solving capabilities. These advanced systems will be cooperative problem-solving systems that effectively combine the different problem-solving skills of humans and computers. A second assumption is that diagnostic systems will be just one component of an integrated maintenance system and will combine job aiding, on-the-job training (OJT), personnel management, and logistics management.

This chapter is organized as follows: (a) examples of maintenance systems which vary the allocation of components of the maintenance task between human and machine, (b) a comparison of human and machine problem solving strengths and weaknesses, and (c) the major research issues in which progress will help make more effective use of human resources.

Examples of Advanced Maintenance Systems

Four hypothetical examples of advanced maintenance systems for equipment diagnosis are presented below. These examples range from a completely automated system to a cooperative human-computer problem-solving system for troubleshooting that incorporates training functions. The purpose of these examples is to show how psychological issues and the state-of-the-art in the areas of BIT, ATE, and AI interact and to further illustrate issues raised in the SSgt Bayshore scenario (Chapter II).

A Completely Automated System

This system makes the strongest assumptions about BIT and ATE technology. It assumes that the maintenance process is totally automated and

that the human is employed only as a sensor and low-level manipulator checking test points specified by the diagnostic programs, inputting readings and other relevant information, and carrying out repairs on instructions from the computer. This first system has greater diagnostic capabilities than the system in the SSgt Bayshore scenario in that it can diagnose all malfunctions without re uir ng any expert human intervention.

For example, in the DELTA system (Bonissone & Johnson, 1984), the system controls the sequence of diagnostic reasoning and involves the technician only when it needs something done. This is a "how-to" information retrieval system added to an AI rule-based diagnostic system. This "how-to" feature is essential to make the team, a low-skill technician and an expert system, productive.

The completely automated scenario has serious personnel and training implications. Such a system would employ personnel with low levels of intellectual ability and skill. These individuals would have to be trained to carry out the various manipulations required by the automated system and to perform various maintenance procedures under system direction. Some procedures are so complex that it may be beyond the capability of the system to instruct an untrained person to perform them. The training necessary to carry out these procedures could present a major problem.

The more serious implication is in the area of morale. Such systems would block acquisition of higher levels of expertise because they would provide no training and the human would be a passive element simply carrying out various kinds of physical manipulations under the directions of computers. Serious morale problems would develop because serving as sensor and manipulator to an automated diagnostic computer would be a low status, unrewarding, dead-end job.

It is an open issue whether the advancing state-of-the-art in AI, ATE, and BIT systems will permit the development of a completely automated system by the early 1990s. A more reasonable assumption is that automated systems will be able to solve a high percentage of routine diagnostic and maintenance problems, but more difficult malfunctions will be corrected by human experts.

An Automated System with Human Experts

This second example of a maintenance system uses low-skilled personnel as sensors and remote manipulators for a large majority of routine fault isolation and correction tasks, but is capable of calling for expert help when necessary.

This example makes strong new assumptions about the state-of-the-art in AI. First, it assumes that the computer is capable of recognizing that it cannot find a solution to a problem and that it must call on expert human assistance. Second, this system has to have the capability of briefing the expert on the current state of a troubleshooting task. Currently, BIT and ATE technology can fail to isolate a fault and provides little or no information to the human expert who is forced to troubleshoot difficult faults with little or no automated assistance.

A system that provides advanced explanations raises issues for the technology of artificial intelligence and an interesting set of psychological questions. What would constitute an adequate explanation for a human expert? How should the automated system present information obtained in the process of attempting fault isolation? Both advanced explanation subsystems and the ability to reason about limitations will require significant advances in the state-of-the-art in AI.

An automated system with human experts also has important personnel and training implications. Such systems require both low-level personnel with the abilities presupposed in the above completely automated scenario and expert personnel who take over when the system cannot succeed in isolating and repairing a malfunction. In addition to the negative implications for low-level personnel, there are also problems with the experts. How are they going to acquire their expertise? A high performance system of the type hypothesized would require very high levels of skill since the system would fail to correct only the most difficult malfunctions.

A Master-Apprentice System

A master-apprentice system is either of the above examples with an integrated training sul system. This scenario makes similar assumptions about ATE, BIT, and AI technology as in the two preceding examples and assumes a well developed, intelligent tutoring system (ITS) technology and the capability of integrating job performance aids with our combined diagnostic-ITS system. These additional capabilities are not an unreasonable extrapolation in the state-of-the-art in AI given that ITS has been an active area of research for many years and that the systems in the above examples have the capability of debriefing human experts.

A master-apprentice system has very favorable personnel and training implications (Denney, Partridge, & Williams, 1983). Although capable of treating a human at the low level of manipulator and sensor, the ITS subsystem would enable the total system to modify its interaction with a technician as that individual advanced in skill level. This system would not intervene in tasks that the human operator had mastered. The objective of such a system would be to train those individuals with the necessary background and intellectual abilities to become expert-level diagnosticians who would take over when the system failed.

An important issue for such master-apprentice systems is coordinating the need to provide training with the need to provide job performance aiding. This is not an easy question. First, job conditions must be such that there is the latitude and flexibility to permit training activities to be going on concurrently with normal productivity. On the flight line, these conditions may never exist or exist only at certain times. This would require analysis. At the intermediate level shop these conditions are perhaps easier to arrange. When it is not possible to permit production and training to go on concurrently, the master-apprentice system is not useful. In this case, however, the production environment could be simulated and the system used in the simulated task environment would be useful as a training aid. This is exemplified by the MAZE in the SSgt Bayshore scenario.

A second issue is that there are difficult questions regarding the sequencing of training and the gradual removal of guidance in a master-apprentice system. Clearly, the master has to know what the apprentice knows and what new subtasks are safe for the apprentice to try. This process could be guided by valid task analyses and the curriculum sequencing methodology of the Instructional Systems Development process.

The first three examples make similar assumptions about technologies being developed to the point where systems can troubleshoot and correct a large majority of equipment malfunctions. However, careless application of this technology could have serious impacts on training of needed experts and could negatively affect morale. However, if a high performance ITS system is incorporated into the state-of-the-art diagnostic system, the negative effects in the areas of personnel and training are ameliorated. This is the reason that SSgt Bayshore enjoys her job.

Mixed-Initiative Human-Computer Diagnostic System

This final example, a mixed-initiative human-computer diagnostic system, exploits the complementary capabilities of the human and computer agents. The purpose is to have the person in the loop, directly involved in diagnostic problem solving cooperatively with ATE or BIT. Human and machine, in this scenario, are working as partners, trying to solve thorny troubleshooting problems beyond the scope of either. Such a system could incorporate ITS capabilities and would require important advances in human-computer problem solving, explanation subsystems, and AI.

The mixed-initiative system would have the same favorable personnel and training implications as those of the preceding example. It would also probably be the most robust and effective of the diagnostic systems in that its problem-solving capabilities would be an effective combination of the complementary capabilities of human and machine.

Comparison of Human and Machine Strengths and Weaknesses

Identifying the most advantageous allocation of maintenance tasks between humans and machines is an important topic. In all four examples of advanced maintenance systems the human was an important component. What was being varied was not the presence of humans, but the allocation of different tasks to either human or computer. The starting point for such allocation decisions is a realistic assessment of the capabilities of humans and the near-term state of AI technology.

Strengths and Weaknesses of the Human as a Problem Solver and Diagnostician

There is general agreement about these strengths and weaknesses. Human beings can be characterized as information processing systems with

computational architecture and capabilities. The strengths of the human problem solver include the following:

- processing of sensory data
- pattern recognition
- skilled physical manipulation but limited physical strength
- some metacognitive skills, e.g., ability to reason about limits of knowledge and skill
- slow but powerful general learning mechanisms
- a large, content-addressable permanent memory
- limited but flexible general problem-solving skills

The weaknesses of the human problem solver are as follows:

- very limited working memory
- limited capability to integrate a large number of separate facts
- tendency to stick with favorite strategies, faults, ways of learning, and preconceptions about the use of tools
- very limited induction capabilities
- lack of consistency
- limitations in the ability to effectively use new information
- emotional and motivational problems
- limited endurance

At some level, all four of the example systems exploit the human's sensory processing, pattern recognition, and manipulation skills. Two major objectives of the research in this area are (a) to design systems that effectively utilize other higher level functions/strengths of the human information processing system and (b) to actively compensate for human limitations.

Strengths and Weaknesses of the Computer

The list of the limitations of the machine component of a human-computer system is an evaluation of the current state-of-the-art in BIT,

ATE, and AI technology and, therefore, subject to revision. Any limitations listed are candidates for active research programs. The strengths of the computer component of the system include the following:

- large processing capacity
- large working memory
- capability of making consistent mechanical inferences taking into account all relevant facts
- capability of processing and utilizing large amounts of actuarial information, e.g., fault histories
- capability to store and retrieve training and reference material
- availability of system is limited only by reliability of basic computer technology
- no emotional or motivational problems

The weaknesses of the computer component of the system are as follows:

- inflexibility
- no or very limited capabilities to adapt to novel situations
- no or very limited learning abilities
- no or very limited metacognitive abilities, i.e., understanding of own limitations
- very difficult programming requirements particularly the current generation of expert systems
- low tolerance for very adverse effects by hostile environment, e.g., rain, loss of power, electro-magnetic pulse

In summary, machines can be surprisingly inflexible diagnosticians and lack common sense reasoning capabilities. A human expert can be an adaptable and effective diagnostician. The primary difficulty is that there is a limited supply of such experts, and a primary motivation for the development of expert systems is to extend the availability of this high-level expertise.

Major Research Issues

In this section six major research issues are addressed that involve the state-of-the-science base necessary for the development of cooperative human-computer, AI-based diagnostic systems and the psychological knowledge necessary to build effective training subsystems.

Models of Diagnostic Problem-Solving Skills

The development of human-computer diagnostic systems requires a detailed understanding of human diagnostic reasoning and problem-solving processes. The very generic understanding of human-computer problem solving that was the basis for the list of strengths and weaknesses listed in the preceding section is not adequate. Such lists are based on knowledge of the general characteristics of the human information processing system, but they do not completely account for the specifics of the problem-solving processes that humans use in various kinds of maintenance tasks.

Explicit models. There are two general models of human diagnostic reasoning (Maxion, 1984; Rouse, 1984): shallow reasoning and deep reasoning. Much human diagnostic problem-solving behavior is mediated by direct associations between symptoms and faults. This is the shallow, symptom-based model since the fault is not inferred from a combination of knowledge of the symptoms and the structure of the unit under test. Rouse claims that this is the most common form of human diagnostic reasoning and is the default mode; experts will only attempt to use more elaborate procedures if pressed by events. Symptom-based diagnostic reasoning is what is captured when a knowledge engineer develops a rule system and encodes the symptom fault relationships known to an expert. The limitations of such reasoning processes are obvious, for they are specific to devices.

The other model of diagnostic reasoning (deep reasoning) involves making inferences about possible faults on the basis of a description of the structure of the device. Deep reasoning is the kind of problem-solving process necessary to deal with a novel device or a novel fault in a known device, in particular interactions between two subsystems which can be very difficult to diagnose. An understanding of the kinds of training that would enable individuals to gain the capability of doing deep reasoning is beginning to develop. The basic cognitive skills and knowledges required are indicated by the AI approaches to specification-based diagnosis described in Chapter III.

Failure modes. Another important aspect involved in developing explicit models is an analysis of failure modes of human diagnostic reasoning and how these modes interact with a specific kind of reasoning process (deep vs. shallow). Extensive analysis of possible failure modes is necessary for the design of cooperative human-computer problem solvers. These types of systems have to be able to make correct inferences about the ongoing problem-solving processes of a partner or student. Detection of a human failure permits the cooperative problem solver system to intervene with an appropriate job aid or information.

The most ubiquitous failure mode, especially for novices, involves loss of information from working memory about intermediate results and goals. The current view of the human information processing system is that the control information that organizes any kind of complex activity is held in working memory which has very limited capacity. Obviously, if critical pieces of control information are lost, it is very difficult for the system to make coherent progress in achieving objectives that are now forgotten.

Authors of works on human-computer problem solving have assumed that the major role of the computer would be to augment the limited working memory by providing a display of current goals and relevant pieces of information. Obviously, this augmentation strategy will be successful only if the information being presented to the human problem solver is in fact relevant. It must also be suitable for incorporation into the human partner's working memory. Bombarding the human partner with a large amount of irrelevant information could cause loss of relevant information from working memory and thereby disrupt the successful sol' ion of the problem. There has also been very little explicit work demonstrating that providing memory aids improves human diagnostic reasoning.

Two other important failure modes are set and functional fixity. These failure modes occur in situations where humans have limited knowledge about possible alternative courses of action. Set is the tendency to persevere on a given hypothesis or problem-solving strategy even after receiving abundant information that invalidates the hypothesis or strategy. The literature on human problem solving shows that set effects are ubiquitous and powerful. Functional fixity refers to the psychological process in which the problem solver will only consider a single function for a component in a situation. Being unable to consider alternative functions in many situations blocks successful problem-solving activities.

Limited inference-making capability is now another well-understood failure mode of human beings. First, memory limitations prevent them from retaining relevant facts. Second, human beings systematically underweight or ignore negative evidence and tend to focus on confirmation of their current hypothesis. There is also a large literature showing that humans do not make effective probabilistic inferences. They tend to systematically misjudge the relative frequency of various types of past events. They do not effectively integrate current evidence with a priori probabilities of various types of failures.

A final failure mode of human beings is that they have very limited attentional capacities. Developing expertise enables the human problem solver to very efficiently allocate this limited attentional capacity. An expert can deal with a large amount of information by knowledgeably selecting information that is relevant to the particular problem-solving activity at hand. Novices, however, do not have the knowledge necessary to understand what is and what is not relevant, and thus, a large amount of supplementary information may only overwhelm their limited attentional capacities.

In summary, detailed understanding of both the processes by which humans carry out diagnostic reasoning and the failure modes of those processes

are necessary for the development of successful cooperative human-computer problem solvers and ITS systems. The most serious limitations of the human processor cannot be compensated for by simply providing more information about the current state of the problem-solving activity or additional background information. A detailed understanding of the actual diagnostic process that a given human problem solver is using is needed in order to be able to specify precisely the additional information or calculational support which would enhance the problem-solving activity. ITS systems would need detailed knowledge of failure modes to detect the occurrence of these failures and intervene with proper job aiding and instructional manipulations.

The Acquisition of Diagnostic Problem-Solving Skills

The ultimate goal of the training process is to develop personnel who have a strong body of generalized diagnostic problem-solving skills so that they can be very rapidly trained to maintain any given system. The difficulty is that there is very little general understanding of the cognitive skills that underly such broad-ranging expertise. Nor is there any explicit understanding of how such skills are acquired.

Instruction in diagnostic problem-solving skills and the delivery of that instruction raises some general questions. The primary question in training is content. Should the focus be on general background knowledge, or should the focus be on the structure and explicit symptom-fault correspondences for a given system?

General background knowledge includes the following:

- basic electronics
- general diagnostic strategies, e.g., split half
- training on generalized maintenance trainers

Instruction relevant to a specific piece of equipment includes the following:

- structure and operating procedures for a specific system
- instruction on specific diagnostic procedures for a system
- instruction in the use of specific job performance aids for a given system

There is some information on the usefulness of instructing novices in general knowledge and problem-solving skills. Rouse (1984) has found that novices rapidly acquire shallow, symptom-based problem-solving strategies. Industrial experience suggests that symptom-based diagnostic reasoning and the use of specific job performance aids that support such problem-solving strategies can be

taught very rapidly in training courses from 6 to 10 weeks in length. However, the goal is to train individuals who are capable of becoming experts with much more general problem-solving skills, i.e., deep reasoners.

General questions about the delivery of instruction involve the division between various kinds of resident instruction (classroom work and dealing with general background knowledge and problem-solving skills) and on-the-job training (the maintenance of a particular system).

Rouse also claims that general diagnostic problem-solving strategies like split-half techniques exploiting limited knowledge of the device's topology are very difficult to teach in isolation, i.e., in the classroom. These strategies are best learned in the context of a specific system, learned again in the context of a quite different system, and then specific instruction given to enable students to abstract these general strategies. Similar assumptions about the learning process have been reported by Anderson (1982, 1983). The most effective kind of training program these results suggest is a brief introduction followed by extensive specific training on one or more classes of systems. Individuals with a year or two of successful field experience could have advanced training on basic electronics and general problem-solving skills.

In both industrial and military practice, specific diagnostic procedures are taught in apprenticeship-type situations on the job. Intelligent tutoring systems incorporated into AI-based cooperative diagnostic systems could dramatically facilitate the delivery of such on-the-job training. Training and job performance aids would be incorporated into a single system. The same basic technology that supports the development of intelligent job performance aids also supports intelligent tutoring. This is exemplified in the SSgt Bayshore scenario where job aiding and training are provided by the same system.

Intelligent Tutoring Systems

Intelligent tutoring systems (ITS) offer a way of providing OJT during the use of an AI-based diagnostic system. ITS seeks to emulate the professional competence of a good teacher working one-on-one with a student. A general prescription for ITS functionality (Anderson, Boyle, Farrell, & Reiser, 1984) would include:

- model the student and reason about the student's knowledge
- instruct in the context of problem solving
- make the goal structure of a problem transparent to the student
- minimize working memory load
- cut off exploration of wrong paths

• facilitate means-ends analysis over analogy

The main components of an ITS are problem-solving expertise, the student model, and tutorial strategies. That is, at the highest level of program organization, ITS consists of an expert student model and tutor module. A very brief summary of the three ITS components follows.

The expert module. The expert module serves as a model of the desired outcome of instruction. In the context of OJT, this module is precisely the expert system that is the basis of the four human-computer systems described at the beginning of this chapter. The role for the expert module is in solving problems in order to evaluate and critique student solutions. It is desirable to have ITS with "articulate" problem solvers that can explain to students how they reached a solution or how they would like the student to try and reach a solution.

The student module. The objective of the student modeling module is to understand what curricular objectives the student has mastered, and to understand or have representations for the student's evolving competence, including if at all possible, predictable misconceptions and suboptimal approaches. Input to the student model may be derived from numerous sources, including (a) a differential comparison of the student's behavior and the behavior output of the expert module on a given problem or question, (b) explicit information derived from direct questions asked of the student, and (c) historical assumptions based on the student's experience. The student's knowledge can be represented in two basic ways.

In the first way, differences between the output of the expert module and the student's performance are compared in terms of a number of issues determined to be of importance in task performance, for example, maximizing the expected information gain for a proposed measurement. An observable psychologically valid process model of expert performance is not necessary. In the second way, a psychologically valid process model of expert performance is employed. These models are usually represented as a production system. With such a process model, the student can be modeled in two ways: in terms of a subset of the expert process model, that subset which accounts for student performance, or in terms of deviations from the expert process model described in the overly generalized, specialized, or otherwise "buggy" perturbations of the rules in the expert model. To be completely satisfactory, the student model must capture the developmental process as the backward chaining approach typical of "uncompiled" novice competence is refined (through practice, experience, and more knowledge) into the pattern matching, methods application, forward reasoning characteristic of expert performance. To merely represent and use the compiled knowledge of the expert in an ITS is not pedagogically useful, as was demonstrated by Clancey in his experiments with MYCIN (Clancey, 1984).

The tutor module. The third module is the tutor module. It serves two basic functions, the first of which is the generation of problems. This function is closely linked with the student model because curricular decisions involve what the student knows or does not know. This information can be thought of as strategic knowledge. A second function of the tutor module involves tactical

knowledge about when the tutor should intervene in the student's problem solving, what it should say when it does interrupt, and how it should answer student questions with explanations that take the student's current state of knowledge into account. All systems do have a scheme for sequencing problems, and approaches vary. Thus, there is a great deal of procedural knowledge involved in effective instruction. This knowledge comprises the basis for the tutoring strategies module of the ITS.

Summary. ITS to date have been "hand crafted" and have usually focused on a subset of issues, concerns, and components of an intelligent tutoring system. This suggests that practical AI applications must not try to press the state-of-the-art on all fronts. Practical efforts should focus on discrete, well defined, and well understood content areas that are priority training areas, such as computer programming and troubleshooting.

In any expert system, performance is an easier goal to achieve than performance with an explanation capability. This, in turn, is easier to achieve than performance with tutorial explanation. This is because tutorial explanation requires a knowledge of a student's competence. In other words, ITS is one of the harder problems in expert systems technology.

Technical Information and Explanation

The availability, accuracy, and usefulness of technical information is a key ingredient in the utilization of human resources. For many tasks, people need technical information in order to do their jobs.

The volume of technical information is growing exponentially. The services have developed automated systems for producing and updating this technical information to assure that the documentation available in the field accurately reflects the latest revision levels of components of a fielded system. Extensive work is also being done on automated storage and retrieval so a technician can rapidly access relevant information in completing a given task. However, the advent of intelligent diagnostic aids will lessen the importance of printed technical documents. For example, the DELTA system combines information retrieval functions into a job performance aid. If a system requests that a given maintenance procedure be carried out, the technician can ask for help. The help is in the form of relevant training material on how to perform various functions, e.g., adjusting a fuel pump available on videodisc that is interfaced to the system.

In intelligent diagnostic assistants, technical information becomes an integral part of the maintenance system. The voluminous contents of technical documentation (circuit schematics, illustrated parts breakdowns, maintenance procedures, etc.) are directly incorporated within the diagnostic aid. For example, circuit schematics become the dependency networks fundamental to the specification-based diagnostic system. Also, since the diagnostic system generates its own diagnostic procedures as needed, hardcopy versions of these in technical publications are no longer necessary.

Much of the information currently in technical documentation is necessary in developing AI-based diagnostic aids. Therefore, in discussing technical information, the issue is not how to research, develop, publish, and distribute technical information per se, but how to:

- gather and input this information in computer-based diagnostic systems

- output this information from a computer-based diagnostic system to a user when needed

Gathering and inputting was discussed in Chapter III in the context of the development of AI programs for failure detection, isolation, recovery, and prediction. The key point is the efficient interfacing of data through the design, engineering, manufacturing, and logistics support processes.

Outputting the information highlights the inverse process of formulating and formatting responses to user requests for information or explanation. Requests for information (e.g., how to do a repair action, or what is the mean time between failure of a component) can be handled like data base queries in that the data are accessed and presented. System response to requests for explanation (e.g., why did the diagnostic system recommend this test?) involves both the diagnostic system's data and the processes which act upon these data.

Providing adequate explanations is important in many expert systems developed in the medical domain and in the four examples presented in this chapter. Highly automated systems that call for expert assistance must be able to successfully brief humans rather than requiring initiation of the diagnostic processes with limited information. In addition, ITS need a comprehensive explanation facility in order to carry out their instructional functions.

The major difficulty in providing adequate explanations is the lack of detailed understanding of the psychological properties of good explanations for job aiding or combined job aiding-training environments. However, a theory of useful explanations can probably be derived from the highly developed work on the psychology of text comprehension (Kieras, 1984) and a better understanding of learning mechanisms.

Explanation content and the most effective method of presentation obviously depend on the context and the goals of the individual receiving the explanation. An explanation must be relevant in the sense that it provides information necessary to the actual ongoing diagnostic reasoning process. There are three important contexts which have very different requirements for adequate explanations:

- during training

- cooperative problem-solving tasks

- briefing of a human expert after a machine has failed

These contexts define two important issues. First, what is an adequate explanation for users at widely varying levels of expertise and with different goals? Second, what is the relevance of user models in explanation subsystems?

Explanations can be generated in one of two ways. A system can be programmed with the appropriate decision rules to retrieve and present relevant portions of independently generated materials. These materials could include reference documentation, training materials, historical data, and other background information. The concern here is selection, format, and presentation.

The other possibility is to derive explanations from system representations of current problem-solving activity, a user model inferred from the human partner's behavior, or from a generalized knowledge of tutorial and instructional strategies. This raises difficult technical issues involving derivations of explanations from various kinds of internal representations, for example, probabilities, list of fired rules, etc. Although it is difficult to derive explanations from probability distributions of possible faults, a good deal of work has been done deriving explanations from lists of fired rules and from the goal trees of a problem reduction-type problem solver.

The Human-Computer Interface

A focus of the SSgt Bayshore scenario was the interface to the aircraft and the integrated maintenance system. The human-computer interface and the operational environment will be critical in developing successful AI-based diagnostic problem-solving systems. These systems will have to operate in hostile environments and situations where there may be real limitations on the kinds of interactions the human can carry out with the system. The technicians also have to communicate and receive information without interrupting their own problem-solving activities.

The details of the interface may be a primary determinant of operational success. The theoretical base and technology for human-computer interface design is well developed and based in part on human factors research. Although it is true that various aspects of the human-computer interface are routinely bungled in the design of new systems, it is not lack of basic knowledge but lack of will to apply this knowledge that leads to these errors. Although not well understood, it is possible that there are specific interface requirements for a cooperative human-computer problem-solving system.

Input. Unconstrained spoken language, constrained voice command, and manual input are three primary ways for technicians to provide information to intelligent diagnostic aids. Voice input is an especially attractive input modality because technicians will often have both hands occupied. Unconstrained continuous spoken language is one of the most difficult AI problems and the research in this area is nowhere near the maturity needed to foster practical results. Fortunately, careful analysis of the dialogue structure and the requirements of the maintenance task would probably indicate that continuous spoken discourse is not necessary.

A "hands free" means of communication with the aid is still desirable, however. Automated recognition of spoken commands is a solution in this case. Off-the-shelf technology exists for this, and vocabularies ranging from tens to hundreds of words can be supported. Modest advances are required, however, to develop speech recognition systems that do not need to be carefully tuned to individual speaker's voice characteristics and to develop systems that can operate in noisy environments. Until such systems become available, all input will need to be through manual means such as keyboards, light pens, mice, touch panels, etc. Often these means will be preferable to spoken commands, indicating that, in general, advanced systems will require a rich variety of input modalities that will be used in different situations.

Output. Naturally, technicians need to be physically involved in their tasks, and good human factors design seeks to free technicians from having to remove hands or eyes from their work. This suggests that both visual and auditory signals should be supplied to the technician in a light, wearable headset, such as the Voice Interactive Maintenance Aiding Device (VIMADS) developed by Honeywell. In VIMADS, voice instructions are provided via earphones, and visual displays are projected on a visor through which the technician can also see. Perfecting this sort of system is mainly a matter of design and packaging. The state-of-the-art in video processing and speech generation is amply developed to support this kind of output device. More difficult is the design of the spoken or visual messages themselves, that is, decisions regarding what information should be provided and how it should be formatted.

In summary, in the cases of both input and output, the major issue is not device technology, but the structure of human-computer dialogues.

Organizational Issues

The likely organizational impact of intelligent maintenance aids in training and on-the-job environments is that organizations will be able to employ a two-tiered approach to training and job design. In the lower tier, intelligent maintenance aids, working in conjunction with unskilled human labor will perform the vast majority of maintenance activities. In the upper tier, maintenance activities that require a degree of technical know-how and sophistication beyond the capability of intelligent maintenance aids will require highly skilled human labor. Organizations will have to develop strategies for sustaining this bi-modal distribution of personnel skills. One strategy is to develop the upper tier from members of the lower tier who show promise for advanced training. A second strategy is to recruit for and maintain each tier separately. Each alternative has recruiting, training, job design, and aiding implications.

Separate tiers. Suppose an organization decides to maintain separate careers for skilled and unskilled maintenance personnel. Consider the unskilled tier. These workers will require training prior to job entry that focuses on overall job orientation and familiarization with the maintenance aid. Detailed technical preparation will not be necessary since the technician will be a sensory and manipulative agent following the directions of the aid. As previously noted,

morale and motivation for workers in this tier may be a problem. This can be mitigated against by recruiting personnel who are not capable of advancement and do not desire opportunity. Turnover would be high because the organization would be making a minimal investment in the worker and the worker will feel a minimal commitment toward the organization.

The upper tier will require personnel with high aptitude. Also required will be extensive training prior to job assignment and continued training on-the-job. In training these personnel, such a tremendous investment will be made that the organization must guard against the premature loss of these assets. This may be accommodated by longer tours of duty for recruits entering this career and a competitive pay scale. Morale and motivation in this population should be high if high expectations are encouraged and opportunity for advancement created. Most probably, all members of the upper tier will play an active role in providing feedback from the field to the agency responsible for maintenance aid development and performance. Some senior members of this upper tier may become the subject-matter experts developing and improving the knowledge base of the maintenance aid itself.

The human factors engineering requirements for a maintenance aid working in a separately tiered organizational strategy are such that "how-to" explanations but not "why" explanations are required when working with unskilled workers. With skilled workers both are necessary. Therefore, the same aid must have both capabilities, and be able to use them selectively. Additionally, the aid must know when it is not successfully completing a problem and be able to support continued learning for upper tier personnel.

Pipelined tiers. Suppose an organization decides to develop the needed skills distribution by "pipelining" selected personnel from the lower tier to the upper tier. Consider the unskilled tier. Since skilled personnel will be drawn from the unskilled labor pool, recruitment must seek to place some people with high aptitude in the lower tier. The expectations of new hires should not be low. People should be informed of their opportunity for advancement, means should be provided to support this transition, and means for the organization to select lower tier candidates for upgrade training. For the lower tier, the training requirements prior to job placement would be the same as in the separate tier scenario. However, training requirements on the job would be different since there is no longer the expectation that people will remain unskilled. On-the-job training opportunities must be provided to enable ambitious personnel to begin skills development.

The implication for the maintenance aid in the pipeline approach is that it act as coach as well as an aid, in the master/apprentice paradigm. Skills development would not be haphazard. Rather, the coach would have to contain a carefully developed curriculum through which it manages worker skills development. This should be done opportunistically, that is, in the context of on-the-job maintenance activity. For example, the coach may begin a dialog with the worker regarding the top-level goal structure of a particular maintenance procedure currently being performed, gradually building up within the worker the capability to understand, remember, and justify each of the steps in the

procedure. The coach would model the skills development of its apprentice, and this model would serve as the organization's selection device in drawing personnel from this tier for advanced training.

Personnel for the upper tier would be selected on the basis of achieving a criterion level of skills development. Since these people had little or no technical training prior to placement in the lower tier, their upgrade training will now provide this background. The specific nature of their training will vary, depending on the organization's intent to use them as specialists or generalists in the upper tier. Considerations of retention, morale, and advancement for the selected personnel would be the same as for the separate tier approach. On-the-job training using the intelligent coach would be continued in the upper tier. Upgrade training would probably focus on basic principles and problem-solving strategies. The intelligent aids would contain a wealth of system-specific information upper tier workers have not encountered. This information would be continually transferred to the highly skilled technicians as assignments bring them into contact with specific equipments.

Implications. The advent of intelligent maintenance aids will not eliminate the need for trained technical personnel. For the foreseeable future, there will always be problems intelligent computer systems cannot solve and which therefore require human intervention. Competent personnel will also be needed where automated systems are unavailable, for example, due to malfunction or power loss. Additionally, trained personnel will be required to provide feedback from the field to designers of intelligent maintenance aids regarding the adequacy of performance. Judgements of this type require technical sophistication.

While not eliminating the need for trained personnel, intelligent aids will probably reduce this need while increasing the opportunity to use unskilled or semi-skilled labor. The issue is how to sustain the resultant bi-modal distribution of skills. The two different approaches offered, separate tiers and pipelining, involve different treatments of recruitment, training, and job design. However, when each scenario is examined across both tiers, the resultant technical requirements for the intelligent aids are substantially the same. In each scenario, the aid must be able to provide lesser skilled personnel "how-to" explanations. In each scenario, the aid must stop work on a problem when the problem lies beyond its competency and provide a useful summary debriefing of the problem-solving activity that it has performed up to the current point. In each scenario, the aid must be able to coach its user. In the separate tier scenario, this coaching is used by the upper tier only, while in the pipeline scenario, it is employed in both tiers.

The human factors engineering features of an intelligent maintenance aid are most likely identical for either scenario. Therefore, the choice between the two scenarios is independent of the aid and supporting artificial intelligence technology. The choice rests on an analysis of organizational values, constraints, resources, and mission.

If all organizational constraints were equal, the pipelining approach would be preferable to the separate tiers approach. First, the pipelining approach

does not suffer from the negative morale and motivation problems which will affect lower tier personnel in the separate tiers approach. Second, the separate tiers approach does not fully utilize all of the required human factors engineering features of the aid. While a coaching capability is needed for the upper tier, and therefore exists within the aid, it is not utilized with the lower tier.

Two subjective reasons also argue for the pipelining approach. The services do have career ladders, that is, sequences of positions, each of which requires slightly more advanced skill and experience. Promotion of personnel through career ladders is currently supported within the job environment. Therefore, the pipelining choice would seem to be the most natural.

The final argument is a humanistic one. While people are capable of sensing and manipulating things, they are also capable of thought. To create a job that does not recognize the potential for reasoned action invites not only sabotage and disrespect, but deprives the organization of the benefits of human diagnostic skill.

V. THE LARGER CONTEXT OF MAINTENANCE

The maintenance problems facing the services today have already been enumerated. Rapid advances in technology, personnel trends, and the dynamic operational scenarios of the future all suggest that these problems will be aggravated in the years to come. Many of the proposed solutions that were offered during the Joint Services Workshop (AFHRL, 1984) share a common theme: the need for an integrated approach to maintenance that encompasses a number of disciplines. In large part, that approach is a matter of logistics.

The Maintenance System

In the narrowest sense, maintenance refers to specific instances of preventive or corrective action applied to specific pieces of equipment. Within this context, logistics can be defined as the planning, allocation, coordination, and support of these actions. In the broadest sense, the maintenance system is comprised of personnel, materiel, facilities, and other related nonmilitary elements. The contribution of logistics to operational readiness is made when these elements are represented in an informational format. Thus, logistics can be thought of as primarily a matter of managing maintenance-related data.

To provide a framework for this discussion, Figure 1 illustrates a fairly typical maintenance system. Rectangles represent the three possible equipment environments: the factory, operations, and maintenance. Broad arrows depict the movement of equipment (including BIT and associated ATE) within the system. That is, equipment is designed and produced by the factory for the field where it is operated and maintained. Although the arrangement of rectangles in Figure 1 implies that equipment alternates between separate operational and maintenance environments, this is not always the case. For aircraft systems, operations and maintenance are relatively separate; for certain shipboard equipment, they are not. The drum-shaped designs indicate relevant on- and off-line data bases and the smaller arrows illustrate the flow of information throughout the system. For instance, the factory supplies schematics, manuals, preventive maintenance schedules, and other reference materials for use in maintenance; the maintenance environment, in turn, relies on a number of additional data bases as well as operations debriefings to keep the equipment mission ready.

Although Figure 1 gives some idea of the complex nature of the logistics task, three additional dimensions are necessary to fully represent the scope of maintenance logistics. First, the relationships among different pieces of equipment must be considered. The logistics associated with a single type of hardware, or even a family of hardware, is costly, but fairly easy to manage. The situation becomes increasingly complex, however, as the variety of equipment increases. Despite economies of scale, competing demands are made on personnel, facilities, time, and inventory. Logistics must coordinate these demands and make optimal use of information that may be generalizeable across types of equipment.

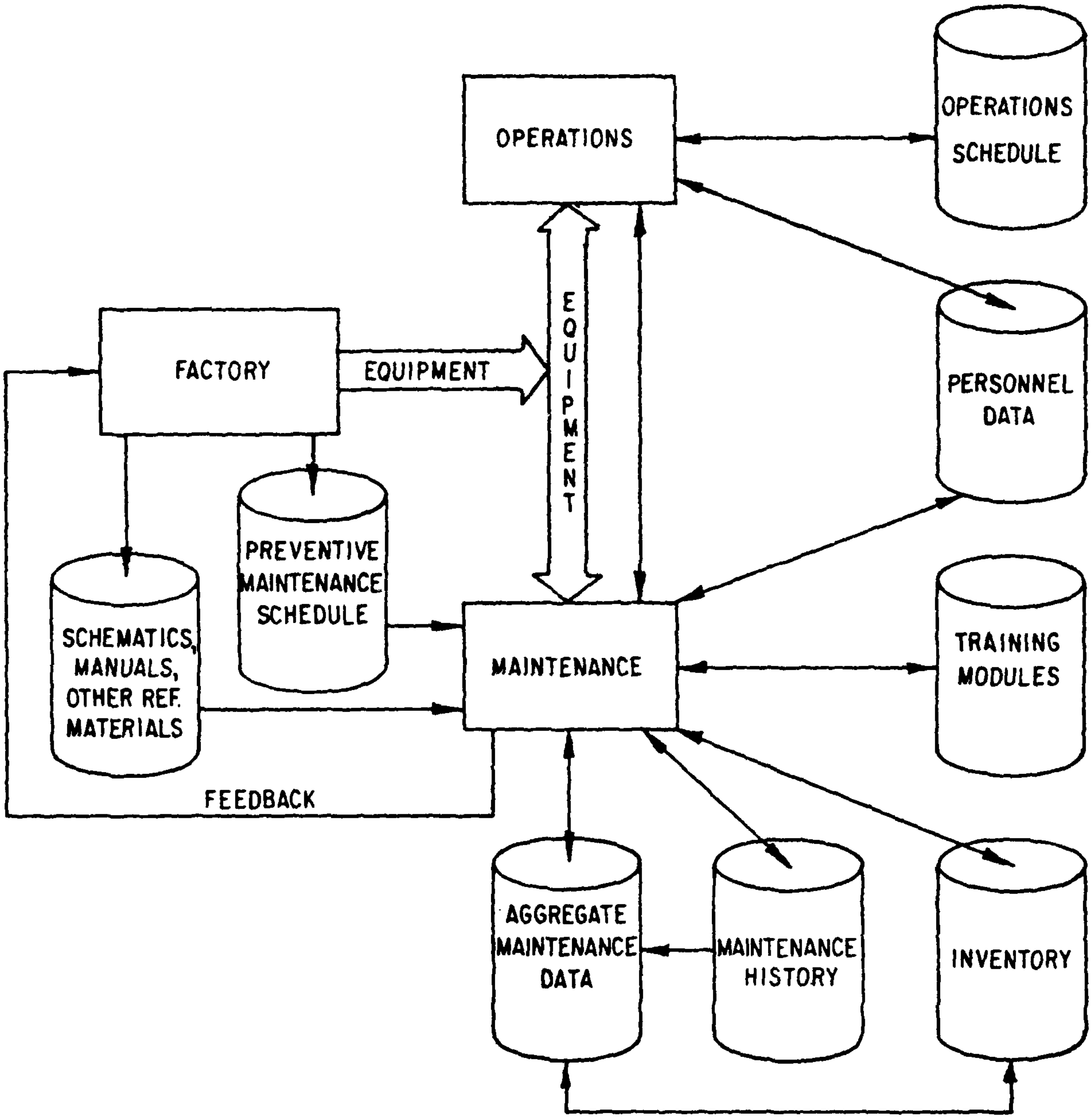


Figure 1. The larger context of maintenance.

Second, maintenance is conducted at various levels within command echelons and functional organizations that differ in terms of their information needs. At the command levels, for example, sorting, filtering, and processing of abundant information is a priority; at the lower levels, the focus is often on augmenting and enhancing scarce information. The three-tier functional approach to maintenance also results in organizational levels with varying needs and goals. Efforts at all levels must be coordinated for the maintenance system to operate efficiently.

Finally, the maintenance system is dynamic. In one sense, the time dimension is represented in Figure 1 by schedules for operations and maintenance. Data bases also change over time as a result of maintenance activities. Time also has implications for logistics because at any one point, different pieces of equipment are at various stages of the equipment life cycle. For example, there are more degrees of freedom in dealing with equipment that is still in its design stages than there are methods of supporting existing equipment.

The following discussion is not a comprehensive review of logistics or potential AI applications to logistics. Its purpose is simply to underscore the importance of logistics considerations for effective maintenance and suggest ways that artificial intelligence techniques could enhance overall system performance by integrating the various elements within the entire maintenance context. The discussion is organized around three basic logistics functions: information management and retrieval, planning and control, and resource allocation.

Information Management and Retrieval

The SSgt Bayshore scenario in Chapter II illustrates how critical the role of information management and retrieval is to future maintenance systems. In this scenario, technical information from a variety of integrated data bases is accessed and manipulated in a user-friendly fashion on the job. This ideal has been the catalyst for a number of projects, such as the Air Force Integrated Maintenance Information System (Dallman, 1984; Johnson, 1981) and the User Defined Technical Information System described by Smillie (1984) at the Joint Services Workshop. However, the realization of this ideal is predicated on changes in the scope and structure of existing information networks and in the nature of knowledge acquisition and retrieval.

Changing the Information Network

Paul Gross, in his Joint Services Workshop address, described a hypothetical shipboard situation in which a surface radar malfunctions (Chapter II). Throughout the course of the Petty Officer Today scenario, information is generated that is not fed back up the line. By expanding and reconfiguring maintenance information networks such as the one shown in Figure 1, that category of information loss can be minimized. Ongoing work using this approach includes the Malfunction Detection, Analysis, and Recording (MADAR) system and

the larger Aircraft Maintenance System (AMS) developed at Dover AFB to support C-5A maintenance. The MADAR/AMS utilizes a variety of interactive data bases to integrate aircraft, component fault history, personnel, job schedule, and parts information.

Other information not always available to the technician concerns the operational environment at the time the fault was detected. Domain-dependent faults are particularly difficult to diagnose when operational conditions cannot be duplicated for the technician.

In some respects, changes in the information network are not necessarily problems for AI. Interfacing various on-line data bases within the system can be accomplished using conventional techniques. The most important interfacing, from an AI perspective, involves the different users. The importance of user-friendly access to information through such techniques as natural language understanding and explanation based on a model of the user gives an AI flavor to conventional systems.

Knowledge Acquisition

A number of AI techniques have been developed to assist in bringing additional data on-line. In the case of information about the operational environment, data are available from monitors within the equipment or from the equipment operator. An operator is a potentially important source of information, but is not generally knowledgeable about maintenance. Therefore, the best approach to collect pertinent and reliable data ma,)e intelligent on-line interrogation.

Expansion of the maintenance information base also implies that a means of automatically extracting information from other data bases or reference materials will be required to deal with the overwhelming volume of technical information. Griffin's paper (1984) outlines such a method of on-line documentation. As designed, the system will read text, extract key words, and integrate the information into the knowledge base. To meet the needs of logistics, such a system would also have to be generic in nature so that it could be used in a wide range of equipment domains.

Retrieval

As the scope and complexity of maintenance information networks increase, efficient access to appropriate data becomes a primary concern. AI offers a great deal to the retrieval process. First, natural language interfacing can be used to make data more accessible to casual users. The PLANES system, for example, accepts requests typed in English for information from the Navy's Maintenance and Material Management (3-M) data base of aircraft flight and maintenance data (Waltz, 1978).

Second, data systems can respond appropriately when the desired information is a deduction rather than a stored fa ; that is, when inferential retrieval is required (cf. Coppola, 1984). KLAUS (Knowledge-Learning and -Using System) is one current project sponsored by the Defense Advanced Research Projects Agency that can determine what a user intends even when that differs from what the user literally requests.

Third, the system should be able to assist the user to search for information that is not well defined. ALOOP (Associative Loop Memory) is an example of a system that allows for this sort of "intelligently guided browsing" (Griffin, 1984).

Finally, the retrieval system could be capable of anticipating different user needs and adapting with experience. Queries from management personnel are likely to require a broad-based search and preliminary analyses. At the technician level, additional information may supplement the answer to a specific question. In either case, the retrieval process can be guided by a model of the user. One fairly simple method of user modeling is to allow the individual user to define words according to his or her working needs. A number of products are already commercially available that use this approach to automatically tailor requests for information.

Planning and Control

The goal of maintenance is to maximize operational readiness, but at the same time there are needs to maximize efficiency and minimize costs. To meet these objectives, planning and control are used to order the sequence of maintenance actions.

Scheduling

Corrective maintenance is not typically a scheduled activity. While there is often some latitude in the order of a maintenance queue (e.g., related to the severity of the malfunction or availability of spares), the process is roughly first-in, first-out. Other activities within the maintenance system, however, such as operations and preventive maintenance, are scheduled in advance. Models already exist that can guide this scheduling process by providing priority rankings to repairable items and determining quantities in the maintenance queue. MISTR (Management of Items Subject to Repair) is one such model that is being developed for depot level scheduling. Traditionally, scheduling programs apply simple but powerful decision analysis techniques to organize the queue under certain well-defined constraints. When the maintenance specifications are potentially incomplete, inconsistent, or qualitative, a knowledge-based approach may be more appropriate. AI models can be used to supply missing details, resolve inconsistencies, determine available options, and identify prerequisites so that maintenance events are coordinated to maximize equipment availability not only at the shop level, but within the larger context of the Command (Coppola, 1984).

Prediction

By analyzing information from a variety of maintenance domains, it becomes possible to predict certain equipment malfunctions before they occur. With this capability, planning and control of corrective maintenance activities also become possible. The MADAR/AMS system mentioned earlier displays some of this predictive quality. This approach may be especially useful for recognizing and dealing with transient faults.

A more integrated logistics system also has the potential to evaluate various aspects of maintenance performance. For example, aggregate data concerning failure rates or invalid equipment returns can be useful in updating the information gain per unit cost metrics within expert systems for diagnosis, assessing individual or shop performance and identifying training needs. Cognitive models and simulations might also be used to evaluate the maintainability of a particular device (Halff, 1984).

Design

Maintenance tasks, whether they involve automatic testing, expert systems, or manual troubleshooting, can be accomplished more efficiently if they are anticipated from the earliest stages of equipment design. This is one reason behind the unified data base technology being developed by AFHRL. By enhancing the availability of logistics support, baseline, and performance data, researchers hope to significantly increase the consideration of logistics factors throughout the system design process.

Resource Allocation

Most maintenance systems experience some disparity between task requirements and resources. Logistics is charged with minimizing that disparity by allocating resources properly. Resource allocation models that support the decision-making process at all levels are necessary to obtain the best possible readiness capability within procurement and repair lead times. Although this function is related to planning and control, there are some additional considerations for the application of AI.

Personnel

The importance of the team concept to maintenance is gaining recognition in the services. Simply put, this concept refers to the fact that many maintenance jobs are very large (e.g., aircraft engine overhaul) or involve equipment systems that are distributed among a number of locations (e.g., a radar system on board ship). Thus, they must be performed by a team or crew rather than a single individual. AI concepts can be useful to support and coordinate maintenance activities in such distributed environments.

In the Air Force, projects such as CODAP (Comprehensive Occupational Data Analysis Programs) provide analyses of occupational data for updating and evaluating classification structures and developing and validating training programs. As the structure of the maintenance system evolves in response to changing technology and personnel, AI might also have a role in the development of new job descriptions and personnel support patterns to ensure that people with maintenance skills are utilized most effectively.

Robotics

Most robotics applications in industry today are related to material handling. These include loading and unloading machines, feeding parts for automated assembly, and presenting parts for inspection. Although many of these activities could also be conducted in depot or other large-scale maintenance settings, it is questionable whether the costs associated with robotics could be justified in terms of increased precision, speed, or safety at this time. As Coppola (1984) points out, current robotics applications in maintenance are limited to automatic test situations. In the near term, however, possibilities exist for the use of robotics for the more complex tasks of diagnosis and repair. The linkage of robot control/programming systems with computer-aided design and manufacturing (CAD/CAM), and other factory data bases (which is expected within 5 years) should help realize this goal (National Research Council, 1983).

In the more distant future, ambulatory robots are envisioned that could be capable of a wide range of maintenance activities (Coppola, 1984). As these applications are realized, robots will become an increasingly important resource for logistics consideration.

Inventory and Supply Management

High false-indication rates result in a particularly high need for ATE and spare parts. This places a heavy burden on limited inventory resources, especially during deployment (e.g., on board ship). If specific equipment repair histories are analyzed in conjunction with aggregate maintenance data, it should be possible to tailor inventories more closely to anticipated needs.

Research, Development, and Application Framework

McGrath (1984) has summarized the operational requirements expected by the services in the next 20 years. Equipment will be technologically complex but dispersed in small, highly mobile units. The logistical demands of such a scenario are substantial. AI techniques are expected to help cope with these demands by:

- expanding the maintenance information network

- automating knowledge acquisition
- providing user-friendly, intelligent retrieval of information from the maintenance data base
- enhancing scheduling, prediction, and evaluation
- incorporating human and expert system models into equipment design
- improving the allocation of personnel, robotics, and inventory

These efforts call for an integrated, multidisciplinary approach that is sensitive to differing organizational and individual needs, but applicable across a wide range of equipment.

REFERENCES

Air Force Human Resources Laboratory. (1984). Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25). Lowry AFB, CO: Air Force Human Resources Laboratory, Training Systems Division. (NTIS No. AD-A145 349)

Anderson, J. R. (1982). Acquisition of cognitive skill. Psychological Review, 89(4), 369-406.

Anderson, J. R. (1983). The architecture of cognition (Cognitive Science Series No. 5). Cambridge, MA: Harvard University Press.

Anderson, J. R., Boyle, C. F., Farrell, R., & Reiser, B. (1984). Cognitive principles in the design of computer tutors. In Proceedings of the Sixth Annual Conference of the Cognitive Science Society. Boulder, CO: Institute of Cognitive Science and the University of Colorado.

Andre, W. L., & Wong, J. T. (1975). Mathematical structure and optimal diagnostic technique for maintenance analysis. In Proceedings of the 36th Military Operations Research Symposium. Moffett Field, CA: U.S. Army Air Mobility R&D Laboratory, Ames Research Center.

Bobrow, D. G., & Hayes, P. J. (Eds.). (1984). Qualitative reasoning about physical systems (Special volume). Artificial Intelligence, 24(13).

Bobrow, D. G., & Winograd, T. (1977). An overview of KRL, a knowledge representation language. Cognitive Science, 1, 3-46.

Bonissone, P. P., & Johnson, H. E., Jr. (1984). DELTA: An expert system for diesel electric locomotive repair. In Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-A145 349). Lowry AFB, CO: Air Force Human Resources Laboratory, Training Systems Division.

Brown, A. (1977). Qualitative knowledge, causal reasoning, and localization of failures (MIT AI-TR-3620). Cambridge, MA: Massachusetts Institute of Technology.

Brown, A., & Sussman, G. J. (1974). Localization of failures in radio circuits--a study in causal and teleological reasoning (MIT AI Memo No. 319). Cambridge, MA: Massachusetts Institute of Technology.

Brown, J. S., Burton, R., & de Kleer, J. (1982). Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II and III. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems. New York: Academic Press.

Buchanan, B. G. (1982). New research on expert systems. In J. E. Hayes, D. Michie, & Y. H. Pau (Eds.), Machine intelligence: Intelligent systems practice and perspective New York: Halsted Press.

Buchanan, B. G., & Feigenbaum, E. A. (1978). DENDRAL and meta-DENDRAL: Their applications dimension. Artificial Intelligence, 11, 5-24.

Cantone, R. (1984). Model-based probabilistic reasoning for electronics troubleshooting. In Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-A145 349). Lowry AFB, CO: Air Force Human Resources Laboratory, Training Systems Division.

Cantone, R., Lander, B. W., Marrone, M. P., & Gaynor M. W. (1984). IN-ATE/2: Interpretating high-level fault modes. In Proceedings of the First Conference on Artificial Intelligence Applications. IEEE Computer Society, Denver, CO.

Cantone, R., Pipitone, F. J., Lander, W. B., & Marrone, M. P. (1983). Model-based probabilistic reasoning for electronics troubleshooting. In A. Bundy (Ed.), Proceedings of the Eight International Joint Conference on Artifical Intelligence (IJCAI-83). Karlsruhe, West Germany, Los Altos, CA: William Kaufmann.

Chandrasekaran, B. (1983). Towards a taxonomy of problem solving types. The AI Magazine, 4(1), 9-17.

Clancey, W. J. (1984). Classification problem solving. In Proceedings of the National Conference on Artificial Intelligence. American Association for Artifical Intelligence, Austin, TX, Los Altos, CA: William Kaufmann.

Clocksin, W. F., & Lellish, C. S. (1981). Programming in prolog. New York: Springer-Verlag.

Coppola, A. (1984). Artificial intelligence applications to maintenance. In Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-A145 349). Lowry AFB, CO: Air Force Human Resources Laboratory, Training Systems Division.

Cramer, M L., et al. (1982). Logic model analysis and standard maintenance information display system (SMIDS) (USAAVRADCOM-TR-81-D-45). Fort Eutis, VA: U.S. Army Research and Technology Laboratories, Applied Technology Laboratory.

Dallman, B. (1984). AFHRL program for artificial intelligence applications to maintenance and training. In Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-A145 349). Lowry AFB, CO: Air Force Human Resources Laboratory, Training Systems Division.

Davis, R. (1976). Applications of meta-level knowledge to the construction, maintenance and use of large knowledge bases. Unpublished doctoral dissertation, Stanford University.

Davis, R. (1982). Expert systems: Where are we? And where do we go from here? The AI Magazine, 3(2), 3-22.

Davis, R. (1983). Diagnosing via causal reasoning: Paths of interaction and the locality principle. In Proceedings of the National Conference on Artificial Intelligence. American Association for Artifical Intelligence, Washington, DC, Los Altos, CA: William Kaufmann.

Davis, R., Shrobe, H., Hamscher, W. Wieckert, K., Shirley, M., & Polit, S. (1982). Diagnosis based on description of structure and function. In Proceedings of the National Conference on Artificial Intelligence. American Association for Artifical Intelligence, Pittsburgh, PA, Los Altos, CA: William Kaufmann.

Davison, J. (1984). Expert systems in maintenance diagnostics for self-repair of digital flight control systems. In Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-A145 349). Lowry AFB, CO: Air Force Human Resources Laboratory, Training Systems Division.

de Kleer, J. (1979). Causal and teleological reasoning in circuit recognition. Cambridge, MA: Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

Denney, R. O., Partridge, M. J., & Williams, R. B. (1983). Integrated testing and maintenance technologies (AFWAL-TR-83-1183). Wright-Patterson AFB, OH: Air Force Wright Aeronautical Laboratories.

DETEX Systems, Inc. What is LOGMOD? and how it helps you! Villa Park, CA: DETEX Systems, Inc., n.d.

Erman, L. D., London, P. E., & Fickas, S. F. (1981). The design and an example use of HEARSAY-III. In A. Drinan (Ed.), Proceedings of the Seventh International Joint Conference on Artificial Intelligence (IJCAI-81). University of British Columbia, Vancouver, Menlo Park, CA: American Association for Artificial Intelligence.

Fahlman, S. E. (1979). NETL: A system for representing and using real-world knowledge. Cambridge, MA: MIT Press.

Fain, J., Gorlin, D., Hayes-Roth, F., Rosenschein, S., Sowizral, H., & Waterman, D. (1981). The Rosie language reference manual (Tech. Rep. N-1647-ARPA). Santa Monica, CA: Rand Corporation.

Fink, P. K., Lusth, J. C., & Duran, J. W. (1984). A general expert system design for diagnostic problem solving. In Proceedings of the IEEE Workshop on Principles of Knowledge-Based Systems. Institute of Electrical and Electronics Engineers, Denver, CO, Silver Spring, MD: IEEE Computer Society Press.

Forgy, C. L. (1981). OPS5 user's manual (Tech. Rep. CMU-CS-81-135). Pittsburgh, PA: Carnegie-Mellon University, Computer Science Department.

Genesereth, M. R. (1982). Diagnosis using hierarchical design models. In Proceedings of the National Conference on Artificial Intelligence. American Association for Artificial Intelligence, Pittsburgh, PA, Los Altos, CA: William Kaufmann.

Grason, J., & Nagle, A. (1980). *Digital test generation and design for testability.* In Proceedings of the 17th Annual Design Automation Conference. Institute of Electrical and Electronics Engineers, Association for Computing Machinery.

Griffin, A. F. (1984). Intelligent information retrieval from on-line technical documentation. In Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-A145 349). Lowry AFB, CO: Air Force Human Resources Laboratory, Training Systems Division.

Gross, P. (1984). An overview of the Joint Logistic Commanders Automatic Test Equipment Panel. In Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-A145 349). Lowry AFB, CO: Air Force Human Resources Laboratory, Training Systems Division.

Gunning, D. R. (1984). Integrated Maintenance Information System (IMIS). Unpublished document, Air Force Human Resources Laboratory, Logistics and Human Factors Division, Wright-Patterson AFB, OH.

Halff, H. (1984). Overview of training and aiding. In Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-A145 349). Lowry AFB, CO: Air Force Human Resources Laboratory, Training Systems Division.

Hamscher, W., & Davis, R. (1984). Diagnosing circuits with state: An inherently underconstrained problem. In Proceedings of the National Conference on Artifical Intelligence. American Association for Artificial Intelligence, Austin, TX, Los Altos, CA: William Kaufmann.

Hart, P. E. (1980). What's preventing the wide spread use of expert systems? Paper presented at the Expert Systems Workshop, San Diego, CA.

Hart, P. E., & Duda, R. O. (1977). PROSPECTOR--A computer-based consultation system for mineral exploration (Tech. Note No. 155). Menlo Park, CA: SRI International, Artificial Intelligence Center.

Hinchman, J. H., & Morgan, M. C. (1984). Application of artificial intelligence to equipment maintenance. In Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-A145 349). Lowry AFB, CO: Air Force Human Resources Laboratory, Training Systems Division.

Holland, J. H. (1980). Adaptive algorithms for discovering and using general patterns in growing knowledge bases. International Journal of Policy Analysis and Information Systems, 4, 3.

Institute for Defense Analyses. (1981). Built-In-Test Equipment Requirements Workshop (IDA Paper P-1600). Arlington, VA: Author.

Johnson, R. C. (1981). Integrated maintenance information system: An imaginary preview (AFHRL-TP-81-18, AD-A104 224). Wright-Patterson AFB, OH: Air Force Human Resources Laboratory, Logistics and Technical Training Division.

Kieras, D. E. (1984). The psychology of technical devices and technical discourse. In Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-A145 349). Lowry AFB, CO: Air Force Human Resources Laboratory, Training Systems Division.

King, J. J. (1982). Artificial intelligence techniques for device troubleshooting (Computer Science Laboratory Technical Note Series CSL-82-9). Palo Alto, CA: Hewlett Packard.

Lahore, H. (1984). Artificial intelligence applications to testability (RADC-TR-84-203). Griffiss AFB, NY: Rome Air Development Center, Air Force Systems Command.

Laffey, T. J., Perkins, W. A., & Nguyen, T. A. (1984). Reasoning about fault diagnosis with LES. In Proceedings of the First Conference on Artificial Intelligence Applications. IEEE Computer Society, Denver, CO.

Longendorfer, B. (1981). Computer-aided testability analysis of analog circuitry. In AUTOTESTCON '81, Institute of Electrical and Electronics Engineers.

Maxion, R. A. (1984). Artificial intelligence approaches to monitoring system integrity. In Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-A145 349). Lowry AFB, CO: Air Force Human Resources Laboratory, Training Systems Division.

McDermott, D. V. (1976). Flexibility and efficiency in a computer program for designing circuits. Cambridge, MA: Massachusetts Institute of Technology.

McDermott, D. V., & Brooks, R., (1982). ARBY: Diagnosis with Shallow Causal Models. In Proceedings of the National Conference on Artificial Intelligence. American Association for Artificial Intelligence, Pittsburgh, PA, Los Altos, CA: William Kaufmann.

McGrath, M. (1984). The need for improvements in weapon system maintenance: What can AI contribute? In Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-A145 349). Lowry AFB, CO: Air Force Human Resources Laboratory, Training Systems Division.

Michalski, R. S. (1980). Pattern recognition as rule-guided inductive inference. Proceedings on PAMI, Institute of Electrical and Electronics Engineers.

Montague, W. E., & Wulfeck, W. H. (1984). Computer-based instruction: Will it improve instructional quality? Training Technology Journal, 1(2), 4-19.

Mooney, J. B., Jr. (1984). Navy requirements. In Proceedings of the Symposium on R&D for Weapon Support and Logistics, Alexandria, VA: National Security Industrial Association, Automatic Test Committee, Logistics Management Committee.

Moorthy, V. S., & Chandrasekaran, B. (1983). A representation for the functioning of devices that supports compilation of expert problem solving structures. In MEDCOMP '83, Second International Conference on Computation Medicine and Medical Computer Science. Athens, OH: Institute of Electrical and Electronics Engineers.

National Research Council. (1983). Applications of robotics and artificial intelligence to reduce risk and improve effectiveness. Washington, DC: National Academy Press.

National Security Industrial Association, Automatic Test Committee, Logistics Management Committee. (1983). In Proceedings of the Conference on Integrated Diagnostics. Tampa, FL: Author.

National Security Industrial Association, Automatic Test Committee, Logistics Management Committee. (1984a). Proceedings of the Symposium on R&D for Weapon Support and Logistics. Alexandria, VA: Author.

National Security Industrial Association, Automatic Testing Committee, Logistics Management Committee. (1984b). Report for the Department of Defense on the implementation of integrated diagnostics. Washington, DC: Author.

Pipitone, F. (1984). An expert system for electronics troubleshooting based on function and connectivity. In Proceedings of the First Conference on Artificial Intelligence Applications. IEEE Computer Society, Denver, CO.

Pople, H. E., Jr. (1982). *Heuristic methods for imposing structure on ill* structured problems: The structuring of medical diagnosis. In P. Sozolovits (Ed.), Artificial intelligence in medicine. Boulder, CO: West View Press.

Reboh, R. (1981). Knowledge engineering techniques and tools in the PROSPECTOR environment (Tech. Note No. 243). Menlo Park, CA: SRI International, Artificial Intelligence Center.

Reggia, J. A., & Perricone, B. T. (1981). Knowledge-based decision support systems: Development through high-level languages. In *Proceedings of the* 20th Annual Technical Meeting of The Association for Computing Machinery (ACM), Washington, DC.

Rieger, C., et al. (1980). ZMOB: A highly parallel multiprocessor (TR-911). College Park, MD: University of Maryland, Computer Science Department.

Rouse, W. B. (1984). Models of natural intelligence in fault diagnosis tasks: Implications for training and aiding of maintenance personnel. In Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-A145 349). Lowry AFB, CO: Air Force Human Resources Laboratory, Training Systems Division.

Sembugamoorthy, V., & Chandrasekaran, B. (in press). Functional representation of devices and compilation of diagnostic problem solving systems. Cognitive Science.

Shortliffe, E. H. (1976). Computer-based medical consultations: MYCIN. New York: American Elsevier Publishing Company.

Shumaker, R. (1984). NAVAIR's AI program for ATE. In Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-A145 349). Lowry AFB, CO: Air Force Human Resources Laboratory, Training Systems Division.

Simpson, W. R., & Agre, J. R. (1983). Adaptive fault isolation with learning. In AUTOTESTCON '83 (83CH1968-7). Institute of Electrical and Electronics Engineers, Fort Worth, TX, New York: IEEE.

Simpson, W. R. & Balaban, H. S., (1982). The ARINC research system testability and maintenance program (STAMP). In AUTOTESTCON '82. Institute of Electrical and Electronics Engineers, Dayton, OH, New York: IEEE.

Smillie, R. J. (1984). Implications of artificial intelligence for a user defined technical information system. In Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-A145 349). Lowry AFB, CO: Air Force Human Resources Laboratory, Training Systems Division.

Stallman, R. M., & Sussman, G. J. (1977). Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis. Artificial Intelligence, 9, 135-196.

Tanner, M. C., & Bylander, T. (1984). Application of the CSRL language to the design of expert diagnosis systems: The Auto-Mech experience. In Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-A145 349). Lowry AFB, CO: Air Force Human Resources Laboratory, Training Systems Division.

van Melle, W. (1982). A domain independent system that aids in constructing knowledge based consultation programs. Unpublished doctoral dissertation, Stanford University, Computer Science Department.

Waltz, D. L. (1978). An english language question answering system for a large relational database. Communications of the ACM, 21(7), 526-539.

Williams, T., & Hinchman, J. (1983). A knowledge-based maintenance aid for diagnosing faults in electronic equipment. In AUTOTESTCON '83 (83CH1968-7). Institute of Electrical and Electronics Engineers, Forth Worth, TX, New York: IEEE.

Wong, J. T., & Andre, W. L. (1976). Some generic properties of a logic model for analyzing hardware maintenance and design concepts. In Proceedings of the Symposium on Applications of Decision Theory to Problems of Diagnosis and Repair. Moffett Field, CA: U.S. Army Air Mobility R&D Laboratory, Ames Research Center.

Wong, J. T., & Andre, W. L. (1981). Logic analysis of complex systems by characterizing failure phenomena to achieve diagnosis and fault-isolation (NASA Tech. Memo 81291). Moffett Field, CA: National Aeronautics and Space Administration.

☆U.S. GOVERNMENT PRINTING OFFICE: 1985 569 053 20049

END

FILMED

12-85

DTIC